LÊ VĂN DOANH - TRẦN KHẮC TUẤN - LÊ ĐÌNH ANH

101 THUẬT TOÁN VÀ CHƯƠNG TRÌNH BẰNG NGÔN NGỮ C



NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT

LÊ VĂN DOANH TRẦN KHẮC TUẤN LÊ ĐÌNH ANH

101 THUẬT TOÁN VÀ CHƯƠNG TRÌNH BÀI TOÁN KHOA HỌC KỸ THUẬT VÀ KINH TẾ BẰNG NGÔN NGỮ C

In lần thứ 6, có chỉnh sửa



NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT
HÀ NỘI – 2006

Lời giới thiệu

Khác với PASCAL, là ngôn ngữ cấp cao được xây dựng với mục đích để học tập và giảng dạy lập trình, ngôn ngữ C và ngôn ngữ phát triển của nó là C++ được xây dựng xuất phát từ những yêu cầu nội tại của tin học, vì thế C phản ánh những thành tựu mới nhất của phần mềm công nghệ tin học đó là tính cô đọng và rất gọn khi lập trình, có thể thâm nhập vào mọi nguồn dữ liệu và hệ thống. C có nhiều mô hình tổ chức bộ nhớ và nói chung người lập trình có thể biết sự tiến triển của chương trình trong bộ nhớ. C có nhiều toán tử rất mạnh. Chương trình viết bằng ngôn ngữ C có thể môđun hóa và ghép nối dễ dàng. C có tất cả những ưu điểm của ngôn ngữ cấp cao nhưng cũng có những đặc trưng của hợp ngữ...

Vì những ưu điểm kể trên, ngày nay các chương trình lớn đều được viết bằng C hoặc đang được viết lại bằng C. Hệ điều hành WINDOWS, UNIX... đều được viết bằng C. Ngôn ngữ C trở nên bắt buộc đối với các nhà lập trình chuyên nghiệp.

Ở Việt Nam, ngôn ngữ C và C++ đang được giảng dạy trong các khoa Công nghệ tin học, Điện tử và thông tin, Đo lường, Điều khiển... của các trường đại học và chuyên nghiệp. Nhiều sách hướng dẫn lập trình ngôn ngữ C do các tác giả trong nước biên soạn và biên dịch đã được xuất bản. Tuy nhiên vẫn còn thiếu bộ sách có tính chất thực hành về C, nhất là còn thiếu những tư liệu, cẩm nang giúp đỡ cho các nhà lập trình bằng ngôn ngữ C đối với các bài toán khoa học kỹ thuật và kinh tế thường gặp.

Cuốn sách **“ 101 thuật toán và chương trình bài toán khoa học kỹ thuật và kinh tế bằng ngôn ngữ C”** nhằm bổ sung cho chỗ trống này. Thông qua các chương trình mẫu các tác giả mong muốn cung cấp cho bạn đọc những thuật toán và chương trình của nhiều loại bài toán khoa học kỹ thuật và kinh tế thường gặp.

Toàn bộ cuốn sách gồm 101 chương trình chia làm 11 chương.

Chương 1 là 20 chương trình minh họa giúp bạn đọc mới bắt đầu làm quen với ngôn ngữ C. Các chương trình này có thuật toán đơn giản và được giới thiệu để minh họa cho những vấn đề cơ bản của việc lập trình ngôn ngữ C.

Chương 2 gồm 16 chương trình dành cho các bài toán đại số với các số thực và số phức.

Chương 3 gồm 8 chương trình đồ họa các hàm theo tham số, theo tọa độ cực, các mặt trong không gian 3 chiều.

Chương 4 gồm 12 chương trình dành cho các bài toán vectơ, ma trận, hệ phương trình đại số tuyến tính.

Chương 5 gồm 8 chương trình về đa thức và nội suy đa thức.

Chương 6 gồm 4 chương trình tính tích phân xác định và tích phân kép.

Chương 7 gồm 3 chương trình giải các phương trình phi tuyến.

Chương 8 gồm 3 chương trình tìm cực trị các hàm một biến và hai biến.

Chương 9 gồm 4 chương trình giải phương trình vi phân và hệ phương trình vi phân cấp n.

Chương 10 gồm 4 chương trình về các vấn đề xác suất và thống kê.

Chương 11 gồm 15 chương trình xử lý văn bản, quản lý và một số bài toán ứng dụng khác.

Tất cả các chương trình đều được chạy có kết quả bằng TC91. Các chương trình của cuốn sách này đều được in trực tiếp từ máy tính.

Về bố cục cuốn sách này có nhiều nét giống cuốn sách " *101 thuật toán và chương trình bài toán khoa học kỹ thuật và kinh tế ngôn ngữ PASCAL* " của cùng tác giả, xuất bản năm 1993 và đã được tái bản nhiều lần. Chúng tôi cũng được sự động viên và góp ý nhận xét của nhiều bạn đọc. Nhiều vấn đề đã được đưa vào cuốn sách này. Tuy nhiên do trình độ có hạn và nhất là do có ít nguồn tư liệu nên chắc chắn cuốn sách này không tránh khỏi sai sót. Chúng tôi mong được sự chỉ dẫn, góp ý của bạn đọc. Mọi chỉ dẫn, góp ý xin gửi về Khoa Thiết bị điện Trường đại học Bách Khoa Hà Nội. Chúng tôi xin chân thành cảm ơn.

Nếu bạn đọc có yêu cầu về đĩa mềm chứa chương trình cuốn sách này xin liên hệ với khoa Thiết bị điện Trường đại học Bách Khoa Hà Nội - ĐT. 692511 chúng tôi xin cung cấp miễn phí.

Các tác giả

Mục lục

Chuong 2. Đại số

Chương 3. Đồ hoạ

Chương 4. Vectơ, ma trận, hệ phương trình đại số tuyến tính

Chương 5. Đa thức và nội suy đa thức

Chương 6. Tích phân số

Chương 7. Giải phương trình phi tuyến

Chương 8. Tối ưu hóa

Chương 9 Phương trình vi phân

Chương 10. Xác suất và thống kê

Chương 11. Một số chương trình xử lý văn bản, quản lý và một số bài toán ứng dụng

Chương 1

MỘT SỐ CHƯƠNG TRÌNH MINH HỌA CƠ BẢN LẬP TRÌNH NGÔN NGỮ C

Chương này dành cho bạn đọc bắt đầu làm quen với ngôn ngữ C. Thông qua một số chương trình đơn giản nhằm minh họa cho những vấn đề cơ bản của việc lập trình ngôn ngữ C. 20 chương trình trong chương này có thuật toán đơn giản, chúng được xây dựng nhằm minh họa, giải thích việc sử dụng các toán tử chủ yếu của ngôn ngữ C.

Bài số 1 Viết chương trình vào số liệu từ bàn phím và hiện kết quả trên màn hình cho các dữ liệu kiểu nguyên, thực và string.

Bài giải Chương trình CT1 gồm việc khai báo kiểu các dữ liệu kiểu nguyên, thực và một string gọi là ten[10], max là 10 ký tự.

Lưu ý là để vào dữ liệu từ bàn phím trong câu lệnh scanf ta phải dùng địa chỉ của biến &a, &b, &x mà không phải là giá trị của biến a, b, x. Phần điều khiển trong câu lệnh printf và scanf nằm giữa hai dấu nháy kép " và " phải có thứ tự đặc tả phù hợp với kiểu của biến cần đưa vào hoặc hiện màn hình.

Nếu muốn in kết quả trên máy in thì câu lệnh in phải được viết như sau :

fprintf (stdprn, " điều khiển ", danh sách biến cần in) ;

```
/* CT1 Chuong trinh minh hoa vao ra */
#include<stdio.h>
void main()
{
int a; /* Khai bao kieu du lieu */
long int b;
float x;
char ten[10]; /* Khai bao string max 10 ky tu */
printf("\n Vao du lieu tu ban phim ");
printf("\n a = "); scanf("%d",&a);
printf("\n b = "); scanf("%ld",&b);
printf("\n x = "); scanf("%f",&x);
printf("\n Ten :");scanf("%s",ten);
        /* Hien thi ket qua */
printf("\n a :%10d\n b :%10ld\n x :%10.2f",a,b,x) ;
printf("\n Tac gia TURBOC :%s",ten);
getch();
return;
}
```

Ta được kết quả chương trình CT1 như sau :

Vao du lieu tu ban phim

a = - 12

b = 6

x = 16.82

Ten : BORLAND

a :   - 12

b =   6

x =   16.82

Tac gia TURBOC : BORLAND

Bài số 2 Tìm kết quả của chương trình sau đây (CT2)

```
/* CT2 Chuong trinh minh hoa cach hien thi */
#include<stdio.h>
void main()
{
int n = 543; /* Khai bao va khoi dau cac bien */
int p = 5;
float x = 34.5678;
printf("\nA:%d %f\n",n,x);
printf("B:%4d %10f\n",n,x);
printf("C:%2d %3f\n",n,x);
printf("D:%10.3f %10d\n",x,n);
printf("E:%-5d %f\n",n,x);
printf("F:%*d\n",p,n);
printf("G:%*.*f\n",12,5,x);
printf("H:%x    :%8x  :\n",n,n);
printf("I:%o    :%8o  :\n",n,n);
getch();
return;
}
```

Bài giải Ta cần nắm vững ý nghĩa của các ký hiệu sau đây :

Ký hiệu	Ý nghĩa
d	dùng cho số nguyên hệ thập phân
o	dùng cho hệ cơ số 8

x	dùng cho hệ cơ số 16 (từ 0 đến f)
X	dùng cho hệ cơ số 16 (từ 0 đến F)
u	dùng cho số nguyên không dấu (unsigned)
c	dùng cho char (ký tự)
s	dùng cho string (xâu ký tự)
e	số thực dạng khoa học m.nnnexx
E	số thực dạng khoa học m.nnnExx
f	số thực dạng mm.nn
z	tương tự như e hay f tuỳ theo giá trị cần chỉ thị

dấu âm (-) để căn sang bên trái, nếu không sẽ căn sang bên phải.

Kết quả của chương trình CT2 là :

A : 543 34.567799

B :   543   34.567799

C : 543 34.567799

D :   34.568   543

E : 543   34.567799

F :   543

G :   34.56780

H : 21f   :   21f   :

I : 1037   :   1037   :

Bài số 3 Cho các biến kiểu thực và kiểu nguyên. Tìm giá trị gán cho các biến của chương trình CT3.

```
/* CT3 Chuong trinh minh hoa lenh gan */
#include<stdio.h>
```

```
void main()
{
int n = 5,p = 9; /* Khai bao va khoi dau cac bien */
int q1,q2,q3,q4,q5;
float x1,x2,x3,x4;
printf("\n Tim gia tri gan cho cac bien ");
q1 = n<p;
q2 = n == p;
q3 = p %n + p>n;
q4 = n*(p>n ? n :p);
q5 = n*(p<n ? p :p);
x1 = p/n;
x2 = (float)p/n;
x3 = (p +0.5)/n;
x4 = (int)(p+0.5)/n;
/* Hien thi gia tri cac bien */
printf("\n q1 = %d ",q1);
printf("\n q2 = %d ",q2);
printf("\n q3 = %d ",q3);
printf("\n q4 = %d ",q4);
printf("\n q5 = %d ",q5);
printf("\n x1 = %10.3f ",x1);
printf("\n x2 = %10.3f ",x2);
printf("\n x3 = %10.3f ",x3);
printf("\n x4 = %10.3f ",x4);
getch();
return;
}
```

Bài giải Ta được kết quả chương trình CT3 là :

Tim gia tri gan cho cac bien

q1 = 1

q2 = 0

q3 = 1

q4 = 25

q5 = 45

x1 = 1.000

x2 = 1.800

x3 = 1.900

x4 = 1.000

Bài số 4 Tìm kết qủa chương trình minh họa toán tử tự tăng tự giảm CT4.

```
/* CT4 Chuong trinh minh hoa toan tu tu tang,tu giam */
#include<stdio.h>
void main()
{
int n = 10, p = 5,q = 10, r;
printf("\n Chuong trinh minh hoa toan tu tu tang tu giam \n") ;
r=n== (p=q);
printf(" A :n =%d p =%d q =%d r =%d\n",n,p,q,r);
n=p=q=5;
n+=p+=q;
printf(" B:n =%d p =%d q =%d \n",n,p,q);
q=n<p?n++ :p++;
```

```
printf(" C :n =%d p =%d q =%d \n",n,p,q)
q=n>p?n++:p++;
printf(" D:n =%d p =%d q =%d \n",n,p,q);
getch();
return;
}
```

Bài giải Ta được kết quả chương trình CT4 là :

Chuong trinh minh hoa toan tu tu tang tu giam
A : n = 10 p = 10 q = 10 r = 1
B : n = 15 p = 10 q = 5
C : n = 15 p = 11 q = 10
D : n = 16 p = 11 q = 15

Bài số 5 Tìm kết quả của chương trình toán tử logic.

```
/* CT5 Chuong trinh minh hoa toan tu logic */
#include<stdio.h>
void main()
{
int n,p,q;
printf(" \n Chuong trinh minh hoa toan tu logic \n ");
n = 5; p = 2; /* Truong hop 1 */
q = (n++ >p )|| ( p++ != 3);
printf("A : n = %d p = %d q = %d\n",n,p,q);
n = 5; p = 2; /* Truong hop 2 */
q = (n++ <p )|| ( p++ != 3);
printf(" B : n = %d p = %d q = %d\n",n,p,q);
```

```
n = 5; p = 2; /* Truong hop 3 */
q = (++n == 3)&&( ++p == 3);
printf(" C : n = %d p = %d q = %d\n",n,p,q);
n = 5; p = 2; /* Truong hop 4 */
q = (++n == 6)&& ( ++p == 3);
printf(" D : n = %d p = %d q = %d\n",n,p,q);
getch();
return;
}
```

Bài giải Toán tử && và || chỉ tính toán hạng thứ hai khi cần. Ở đây trường hợp 1 và 3 không được tính. Kết quả chương trình CT5 là:

Chuong trinh minh hoa toan tu logic

A : n = 6 p = 2 q = 1

B : n = 6 p = 3 q = 1

C : n = 6 p = 2 q = 0

D : n = 6 p = 3 q = 1

Bài số 6 Cho mảng 3 phần tử kiểu nguyên và một con trỏ kiểu nguyên. Tìm kết quả của chương trình CT6.

```
/* CT6 Chuong trinh minh hoa bang va con tro */
#include<stdio.h>

void main()
{
int i,j;
int t[3]; /* Khai bao mang 3 phan tu */
int * adt; /* Khai bao con tro kieu nguyen */
```

```
for (i = 0,j = 0; i < 3; i++) t[i] = j++ + i ; /* 1 */
printf("\n");
for (i = 0; i < 3 ;i++) printf(" %d", t[i]); /* 2 */
printf("\n");
for (i = 0; i < 3 ;i++) printf(" %d", *(t+i)); /* 3 */
printf("\n");
for (adt=t ;adt<t+3 ;adt++) printf(" %d",* adt) ; /* 4 */
printf("\n");
for (adt=t+2;adt>=t;adt--) printf(" %d",* adt); /* 5 */
getch();
return;
}
```

Bài giải Ta được kết quả của chương trình CT6 như sau :

0 2 4

0 2 4

0 2 4

4 2 0

Bài số 7 Lập chương trình để phân loại học sinh theo điểm.

Bài giải Có nhiều phương án để rẽ nhánh chương trình này. Ở đây ta sử dụng câu lệnh if else để kiểm tra điểm của học sinh và rẽ nhánh chương trình.

Từ 0 đến 3 điểm thuộc loại kém.

4 điểm thuộc loại yếu.

Từ 5 đến 6 điểm thuộc loại trung bình.

Từ 7 đến 8 điểm thuộc loại khá.

Từ 9 đến 10 điểm thuộc loại giỏi.

Ngoài các điểm từ 0 đến 10 sẽ báo vào sai. Dùng lệnh nhảy với goto (nhãn tt) để tiếp tục chu trình.

```
/* CT7 Chuong trinh minh hoa dung if else */
#include<stdio.h>
void main()
{
int diem;
printf ("\n Chuong trinh phan loai hoc sinh theo diem ");
tt: printf ("\n VAO SO LIEU ");
printf("\n diem = "); scanf("%d",&diem);
if(( diem >= 0) && (diem <= 3))
    printf(" Kem \n");
    else if( diem == 4)
    printf(" Yeu \n");
        else if(( diem >= 5) && (diem <= 6))
        printf(" Trung binh \n");
            else if( diem >= 7 && diem <= 8)
            printf(" Kha \n");
                else if (( diem >= 9) && (diem <= 10))
                printf(" Gioi \n");
                    else printf(" Vao sai \n") ;
printf("\n tiep tuc 1 / stop 0 : ");
scanf(" %d", &diem);
if ( diem == 1) goto tt;
getch();
return;
}
```

Kết quả chương trình CT7 như sau :

Chương trinh phan loai học sinh theo diem

Vao so lieu

diem = 9

Gioi

tiep tuc 1 / 0 : 0

Bài số 8 Lập chương trình phân loại học sinh theo điểm sử dụng switch.

Bài giải Để phân nhánh chương trình ta sử dụng câu lệnh switch với các nhánh là kết quả điểm của học sinh.

```
/* CT8 Chuong trinh minh hoa su dung switch */
#include<stdio.h>
void main()
{
int diem;
tt : printf(" Vao du lieu \n");
printf("\n diem = ") ; scanf("%d", &diem);
switch(diem)
{
    case 0:
    case 1:
    case 2:
    case 3: printf(" Kem \n ");break;
    case 4: printf(" Yeu \n ");break;
    case 5:
    case 6: printf(" Trung binh \n ");break;
```

```
        case 7:
        case 8: printf(" Kha \n ");break;
        case 9:
        case 10: printf(" Gioi \n");break;
        default: printf(" Vao sai \n");
    }
    printf("\n tiep tuc 1 / stop 0 :
    scanf("%d", &diem);
    if (diem == 1) goto tt;
    getch();
    return;
    }
```

Kết quả chương trình CT8 là :

Vao du lieu

diem = 4

Yeu

tiep tuc 1 / 0: 0

Bài số 9 Tìm tất cả các phương án để có được 1000đ từ các tờ giấy bạc 500đ, 200đ và 100đ.

Bài giải Ta dùng 3 chu trình for lồng vào nhau để thay đổi số tờ giấy bạc 500đ, 200đ và 100đ với điều kiện tổng của chúng bằng 1000đ.

```
/* CT9 Chuong trinh minh hoa su dung cau lenh for */
#include<stdio.h>
void main()
{
int sopa; /* bo dem so phuong an */
```

```
int t50υ, /* tien 500 d */
int t200; /* tien 200 d */
int t100; /* tien 100 d */

sopa = 0;
for ( t500 = 0;, t500 <= 2; t500++)
    for ( t200 = 0; t200 <= 5; t200++)
        for ( t100 = 0; t100 <= 10; t100++)
    if ( 100*t100 + 200*t200 +500*t500 == 1000)
    {
    sopa ++;
    printf("\n Tien 1000 d = ");
    if( t100 ) printf("     %2d X 100 d ",t100);
    if( t200 ) printf("     %2d X 200 d ",t200);
    if( t500 ) printf("     %2d X 500 d ",t500);
    printf("\n");
    }
printf("\n Co tat ca %d phuong an de co 1000 d ",sopa);
getch();
return;
}
```

Kết quả ta được :

Tien 1000d = 10 X 100d

Tien 1000d = 8 X 100d 1 X 200d

Tien 1000d = 6 X 100d 2 X 200d

Tien 1000d = 4 X 100d 3 X 200d

Tien 1000d = 2 X 100d 4 X 200d

Tien 1000d = 5 X 200d

Tien 1000d = 5 X 100d 1 X 500d

Tien 1000d = 3 X 100d 1 X 200d 1 X 500d

Tien 1000d = 1 X 100d 2 X 200d 1 X 500d

Tien 1000d = 2 X 500d

Co tat ca 10 phuong an de co 1000d

Bài số 10 Tìm kết quả của chương trình minh họa sử dụng **While** CT10.

```
/* CT10 Chuong trinh minh hoa su dung while */
#include<stdio.h>
void main()
{
int n,p;
n = 0 ;
printf("\n Chuong trinh minh hoa su dung while \n ");
while ( n <= 5 ) n++;
printf("A : n = %d \n",n);

n = p = 0;
while ( n <= 8 ) n += p++;
printf(" B : n = %d \n",n);
n = p = 0;
while ( n <= 8 ) n += ++p;
printf(" C : n = %d \n",n);
n = p = 0;
while ( p <= 5 ) n += p++;
printf(" D : n = %d \n",n);
n = p = 0;
```

```
while ( p <= 5 ) n += ++p;
printf(" E : n = %d \n",n);
getch();
return;
}
```

Bài giải Kết quả của chương trình CT10 như sau :

A : n = 6

B : n = 10

C : n = 10

D : n = 15

E : n = 21

Bài số 11 Tính tổng n số đầu tiên của cấp số

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \qquad + \frac{1}{n}$$

Bài giải Dùng một chu trình for để tính tổng của n số đầu tiên. Trong đó biểu thức tong + = (float) 1 / i ; được chuyển dạng thành float.

```
/* CT11 Chuong trinh minh hoa su dung do while */
void main()
{
int nt; /* so thanh phan cua chuoi */
float tong; /* tong cua chuoi */
int i;
do
    { printf("\ Cho so thanh phan n = ");
    scanf("%d",&nt);
    }
```

```
while (nt<1);
for ( i = 1,tong = 0;i <= nt; i++) tong += (float)1/i;
printf("\n Tong %d so dau tien = %f",nt,tong);
getch();
return;
}
```

Kết quả của chương trình CT11 là :

Cho so thanh phan n = 5

Tong 5 so dau tien = 2.83334

Nhận xét : Trong 3 thành phần cách nhau bằng dấu ; của câu lệnh for có thể vắng mặt nhiều thành phần hoặc có thể là một dãy lệnh. Ở CT11 trong thành phần thứ nhất của for là i = 1, tong =0;

Bài số 12 Xây dựng hàm để in một xâu ký tự

Bài giải Ta xây dựng 3 hàm :

void ham1 (void) dùng in một string không có đối số và cũng không có giá trị trả về.

void ham2 (int) có đối số kiểu int và không có giá trị trả về và được gọi hai lần.

int ham3 (int) có đối số kiểu int và có giá trị trả về (trả về 0) và được gọi 3 lần.

```
/* CT12 Chuong trinh minh hoa khai bao va goi ham */
#include<stdio.h>
void ham1 (void);
void ham2 (int);
int ham3 (int);
```

```
void main()
{
printf("\n Chuong trinh minh hoa khai bao va goi ham \n");
ham1();
ham2(2); /* 2 lan goi ham */
ham3(3); /* 3 lan goi ham */
getch();
return;
}
void ham1 (void)
    {
    printf( " TURBOC \n");
    }
void ham2 (int n)
    {
    int i ;
    for ( i = 0; i < n; i++)
        printf( " Tac gia ngon ngu C \n");
    }
int ham3 (int n)
    {
    int i;
    for ( i = 0; i < n; i++)
        printf(" Dennis Ritchie \n");
        return 0;
    }
```

Kết quả chương trình CT12 như sau :

Chuong trinh minh hoa khai bao va goi ham

TURBOC

Tac gia ngon ngu C

Tac gia ngon ngu C

Dennis Ritchie

Dennis Ritchie

Dennis Ritchie

Bài số 13 Viết một hàm nhận hai đối số là số thực và một ký hiệu ứng với một trong bốn phép tính số học + - * /

Bài giải Cần xây dựng một hàm gọi là SOHOC (float, float, char) để thực hiện các phép tính số học cộng, trừ, nhân, chia và ghi kết quả.

```
/* CT13 Chuong trinh mimh hoa su dung ham */
#include<stdio.h>
float SOHOC (float ,float ,char); /* prototype */
void main()
{
float x,y;
printf("\n VAO DU LIEU ");
printf("\n x = "); scanf("%f",&x);
printf("\n y = "); scanf("%f",&y);
printf("\n Tong la : %f ", SOHOC(x,y,'+'));
printf("\n Hieu la : %f ", SOHOC(x,y,'-'));
printf("\n Tich la : %f ", SOHOC(x,y,'*'));
printf("\n Thuong la : %f ",SOHOC(x,y,'/'));
getch();
return;
}
```

```
float SOHOC ( float v1,float v2,char tinh ) /* dinh nghia ham */
    {
    float ketqua;
    switch (tinh)
    {
    case '+' : ketqua = v1 + v2;
              break;
    case '-' : ketqua = v1 - v2;
              break;
    case '*' : ketqua = v1 * v2;
              break;
    case '/' : ketqua = v1 / v2;
              break ;
    default : ketqua = v1 + v2;
    }
    return ketqua;
    }
```

Kết quả của chương trình CT13 là :

VAO DU LIEU

x = 12

y = 24.5

Tong la : 36.500000

Hieu la : -12.500000

Tich la : 294.000000

Thuong la : 0.489796

Bài số 14 Cho mảng một chiều, vào dữ liệu mảng từ bàn phím và tính tổng của các phần tử của mảng.

Bài giải Dùng cho chu kỳ for để thay đổi chỉ số phần tử của mảng và vào số liệu phần tử của mảng. Dùng một chu kỳ for để tính tổng các phần tử của mảng.

```
* CT14 Chuong trinh minh hoa mang mot chieu */
#include<stdio.h>
void main()
{
int i;
float m[5],s;
for ( i = 0; i < 4; ++i)
    {
    printf("\n m[%d] = ",i):scanf("%f", &m[i]);
    }
    s = 0;
    for ( i = 0; i <= 4;i++ )
        {
        s = s+ m[i];
        printf("\n m[%d] = %8.2f",i,m[i]);
        }
printf("\n Tong = %8.2f",s);
getch();
return;
}
```

Kết quả chương trình CT14 như sau :

m[0] = -28

m[1] = 51.3

m[2] = 15

m[3] = 87.6

m[4] = 21.2

m[0] = -28.00

m[1] = 51.30

m[2] = 15.00

m[3] = 87.60

m[4] = 21.20

Tong = 147.10

Lưu ý phép toán lấy địa chỉ chỉ áp dụng được đối với các phần tử của mảng 1 chiều.

Bài số 15 Cho mảng một chiều. Vào dữ liệu các phần tử của mảng và tính tổng của các phần tử của mảng, sử dụng con trỏ.

Bài giải Dùng con trỏ pm trỏ tới phần tử của mảng. Dùng chu trình for để vào các phần tử của mảng và tính tổng của các phần tử.

```
/* CT15 Chuong trinh minh hoa mang dung con tro */
#include<stdio.h>
void main()
{
int i ;
float s, *pm, m[5];
pm = m ;
for ( i = 0; i < 5; i++ )
    {
    printf("\n m[%d] = ",i); scanf("%f", &pm[i]);
    }
s = 0;
for ( i = 0; i < 5; i++ )
```

```
    s += m[i];
    printf("\n m[%d] = %8.2f",i,pm[i]);
    }
printf("\n Tong la = %8.2f",s);
getch();
return;
}
```

Kết quả ta được :

m[0] = 45

m[1] = 27

m[2] = 23

m[3] = 67.5

m[4] = 0

m[0] = 45.00

m[1] = 27.00

m[2] = 23.00

m[3] = 67.50

m[4] = 0.00

Tong = 162.50

Bài số 16 Viết chương trình đòi hỏi người sử dụng đưa vào một số nguyên nằm giữa 1 và 7 để hiện màn hình tên ngày trong tuần theo con số đưa vào (1: thứ hai, 2: thứ ba, . . . , 7: chủ nhật).

Bài giải Ta tạo nên một mảng 7 con trỏ về một string ứng với tên mỗi ngày trong tuần. Vì xâu này là hằng nên có thể tạo nên một mảng được khởi đầu có dạng char *ngay [7] = {"thu hai", "thu ba",...}; ngay[0] chứa địa chỉ của xâu có ký tự đầu tiên tức là "thu hai", ngay [1] chứa

địa chỉ xâu ký tự "thu ba" Để hiện màn hình giá trị xâu ký tự i cần lấy địa chỉ ngay [i-1].

```
/* CT16 Chuong trinh minh hoa mang string */
#include<stdio.h>
void main()
{
int i;
char *ngay[7] = { " Thu hai "," Thu ba "," Thu tu ",
    " Thu nam"," Thu sau "," Thu bay "," Chu nhat "};
printf("\n Chuong trinh minh hoa mang string ");
do
    {
    printf("\n Cho mot so nguyen giua 1 va 7 :");scanf("%d",&i);
    }
while ( i <= 0 || i > 7 );
printf("\n Ngay so %d cua tuan la : %s",i,ngay[i-1]);
getch();
return;
}
```

Ta được kết quả chương trình CT16 như sau :

Chương trình minh hoa mang string

Cho mot so nguyen giua 1 va 7 : 5

Ngay so 5 cua tuan la : Thu sau

Bài số 17 Lập chương trình đếm lần xuất hiện một ký tự trong một xâu ký tự.

Bài giải Chương trình cần khai báo một dòng nhiều nhất 128 ký tự và đọc xâu ký tự bằng hàm gets và tổ chức một chu trình while để đếm

số lần xuất hiện của một ký tự (trong CT17 là chữ a).

```
/* CT17 Chuong trinh minh hoa xu ly string */
#include<stdio.h>
#define HANG 128
void main()
{
char dong[HANG]; /* khai bao mot dong 128 ky tu */
int i; /* xet cac ky tu cua dong */
int na; /* dem chu 'a' */
printf("\n Cho mot dong chu : " );
gets (dong);
na= 0;
i = 0;
while(dong[i]) if (dong[i++] == 'a') na++;
printf("\n Dong co %d chu a ",na);
getch();
return;
}
```

Kết quả chương trình CT17 như sau :

Cho mot dong chu : Cong hoa xa hoi chu nghia Viet Nam

Dong co 4 chu a

Bài số 18 Cho một mảng hai chiều, tìm phần tử âm đầu tiên trong mảng.

Bài giải Cần khai báo và khởi đầu mảng và dùng hai chu trình for để thay đổi chỉ số i,j của các phần tử. Dùng if (a[i][j]<0) để tìm phần tử âm.

```
/* CT18 Chuong trinh minh hoa mang nhieu chieu */
#include<stdio.h>
void main()
{
/* Khai bao va khoi dau mang hai chieu */
int i,j
float a[3][4] = {
                { 4 , 2.5,6.2, -8 },
                { 3.2, 25, 7 , 0.5},
                { 1.5,-3 ,0, 5 }};
for ( i = 0; i < 3; i++)
   for ( j = 0; j < 4; j++)
   if ( a[i][j] < 0 ) goto kqua;
      printf("\ Mang khong co phan tu am ");
      goto end;
kqua : printf("\n Phan tu am dau tien la a[%d,%d] = %8.2f",
   i+1,j+1,a[i][j] ) ;
   end:
   ; /* Cau lenh rong */
   getch();
   return;
}
```

Kết quả chương trình CT18 như sau :

Phan tu am dau tien la a[1,4] = -8.00

Bài số 19 Lập một cấu trúc để ghi ngày sinh, ngày tuyển dụng và bậc lương của cán bộ.

Bài giải Tổ chức một cấu trúc gọi là CANBO có trường là ngày sinh, ngày tuyển dụng, bậc lương và một cấu trúc gọi là DATE có trường là ngày tháng năm dùng cho ngày sinh và ngày tuyển dụng.

```
/* CT19 Chuong trinh minh hoa su dung cau truc */
#include<stdio.h>
void main()
{
 struct DATE /* Dinh nghia cau truc thoi gian */
      {           /* Truong cua cau truc */
      int ng;  /* Ngay */
      char t[10]; /* Thang la string nhieu nhat 10 ky tu */
      int nam;   /* Nam */
      };
struct CANBO /* Dinh nghia cau truc nhan su */
    {          /* Truong cua cau truc */
    struct DATE dates; /* Ngay sinh */
    struct DATE datet; /* Ngay tuyen dung */
    float luong;         /* Bac luong */
    };
struct CANBO c;
printf("\n Vao ngay sinh ngay thang nam : ") ;
scanf(" %d%s%d", &c.dates.ng,c.dates.t,&c.dates.nam);
printf("\n Vao ngay tuyen dung ngay thang nam : ");
scanf(" %d%s%d",&c.datet.ng,c.datet.t,&c.datet.nam);
printf("\n Tien luong : "); scanf("%f", &c.luong);
printf("\n%4d%10s%6d",c.dates.ng,c.dates.t,c.dates.nam);
printf("\n%4d%10s%6d",c.datet.ng,c.datet.t,c.datet.nam);
printf("\n Luong = %8.2f",c.luong);
```

```
getch();
return;
}
```

Kết quả chương trình CT19 như sau :

Vao ngay sinh ngay thang nam : 20 tam 1944

Vao ngay tuyen dung ngay thang nam : 15 muoi 1965

Tien luong : 650000

20 tam 1944

15 muoi 1965

Luong = 650000.00

Bài số 20 Tìm chương trình thay thế các ký hiệu của chương trình CT20.

Bài giải Cần thận trọng trong việc dùng macro. Bộ tiền xử lý của chương trình biên dịch chỉ thực hiện xử lý văn bản mà không chú ý đến nội dung. Chương trình thay thế các ký hiệu của CT20 là :

```
void main ()
{
int x,y,z;
x = 5;
y = 10*5;
printf("\n QUA LON");
z = x + y;
z = x + y + 6;
z = 5*x + y;
z = 5 * ( x + y );
z = 5 *( ( x ) + ( y ));
printf("\n z = %d" ,z);
getch();
```

```
return;
{

/* CT20 Chuong trinh minh hoa su dung macro */
#define BEGIN {
#define END }
#define INTEGER int
#define NB 10
#define LIMIT NB*5
#define SUMXY x+y
#define SUM1 (x+y)
#define SUM2 ((x)+(y))
#define MSG "\n QUA LON"
void main()
BEGIN
INTEGER x, y, z;
x=5;
y=LIMIT;
printf(MSG);
z=SUMXY;
z=SUMXY+6;
z=5*SUMXY;
z=5*SUM1;
z=5*SUM2;
printf("\n=%d",z);
getch();
return;
END
```

Chương 2

ĐẠI SỐ

Chương này trình bày một số chương trình thực hiện các phép tính đại số trên các số thực và số phức. Ở đây chỉ giới hạn tronglớp các bài toán đại số thường gặp trong các bài toán kỹ thuật và kinh tế.

Bài số 21 Tính tổ hợp chập k trong n phần tử C_n^k.

Bài giải Tổ hợp chập k trong n phần tử ký hiệu C_n^k được định nghĩa như sau :

$$C_n^k = \frac{n!}{k!(n-k)!}$$

Ta có thể tính tổ hợp C_n^k trực tiếp từ định nghĩa bằng cách tính các giai thừa. Tuy nhiên ta cũng có thể tính tổ hợp đơn giản hơn bằng cách sử dụng công thức truy hồi :

$$C_n^k = \frac{n-k+1}{k} C_n^{k-1} , \qquad \text{với } k = 1,2,...,n$$

Ta tổ chức một chu trình nhằm tính lặp C_n^k theo C_n^{k-1}.

Chương trình gồm việc vào các số liệu sau : số các phần tử n, chập k và một chu trình for để tính tổ hợp.

```
/* CT21 Chuong trinh tinh to hop */
#include <stdio.h>
main()
{
int n,k,i;
float c;

printf("\n Chuong trinh to hop chap k trong n phan tu");
printf("\n VAO SO LIEU \n");
printf(" n = ");scanf("%d",&n);
printf(" k = ");scanf("%d",&k);
c = 1;
for (i=1;i<=k;i++)
    {
    c = (n-i+1)*c/i;
    }
printf("To hop chap %d trong %d phan tu : %10.2f\n",k,n,c);
getch();
return;
}
```

Nếu vào các dữ liệu sau đây thì có kết quả chương trình :

Chuong trinh to hop chap k trong n phan tu
VAO SO LIEU
n = 10
k = 4
To hop chap 4 trong 10 phan tu : 210.00

Bài số 22 Tính tích của một dãy n số thực.

Bài giải Để tính tích của một dãy n số thực ta áp dụng công thức truy hồi :

P = P*x[i] , với giá trị đầu P = 1

Chương trình bao gồm những bước sau đây :

- Đọc dãy n số thực, đây là một mảng n phần tử giá trị thực được đưa vào nhờ câu lệnh for.

- Tổ chức một chu trình tính tích và ghi kết quả.

```
/* CT22 Chuong trinh tinh tich n so thuc */
#include <stdio.h>
main()
{
int i,n;
float tich,x[100];
printf(" Chuong trinh tinh tich cua n so thuc\n");
printf(" VAO SO LIEU\n");
printf("n = ");scanf("%d",&n);
    for (i=0;i<n;i++)
    {
    printf("x[%d] = ",i);
    scanf("%f",&x[i]);
    }
    tich=1;
    for (i=0;i<n;i++)
    {
    tich *=x[i];
    }
```

```
printf(" TICH SO = %10.3f\n ",tich);
getch();
return;
}
```

Trong chương trình này nếu đưa vào mảng dữ liệu như sau ta sẽ được kết quả :

Chuong trinh tinh tich cua n so thuc

VAO SO LIEU

n = 4

x[0] = 12.52

x[1] = -45.6

x[2] = 25

x[3] = 50.45

TICH SO = -720062.750

Bài số 23 Tính tổng và tích của dãy n số thực

Bài giải Thuật toán áp dụng cho bài này cũng tương tự bài 22

Chương trình gồm các bước sau :

- Nhập bảng dữ liệu n phần tử từ bàn phím

- Tổ chức một chu trình for để tính tổng và tính tích của n số thực theo các công thức lặp.

s = s + x[i]

p = p * p[i]

với bước ban đầu s = 0 , p = 1.

```
/* CT23 Chuong trinh tinh tong va tich */
#include <stdio.h>
main()
{
int i,n;
float p,s,e,x[10];

printf(" VAO SO LIEU \n");
printf("n = ");scanf("%d",&n);
for (i=1;i<=n;i++)
    {
    printf("x[%d] = ",i);
    scanf("%f",&x[i]);
    }
s = 0;
p = 1;
for (i=1;i<=n;i++
    {
    p *= x[i];
    s += x[i];
    }
e = p+s;
printf("Ket qua Tich = %10.3f\n",p);
printf("Ket qua Tong = %10.3f\n",s);
printf("Ket qua Tong va Tich = %10.3f\n",e);
getch();
return;
}
```

Trong chương trình CT23 nếu ta vào các dữ liệu từ bàn phím sau đây sẽ có kết quả :

VAO SO LIEU

n = 4

x[1] = 45.5

x[2] = 24.25

x[3] = 15

x[4] = 67.87

Ket qua Tich = 112391.000

Ket qua Tong = 152.602

Ket qua Tong va Tich = 1123443.625

Bài số 24 Tìm các số dương trong một dãy số

Bài giải Chương trình gồm việc vào dãy số từ bàn phím và dùng chu trình for để kiểm tra các giá trị x[i] > 0 và ghi kết quả.

```
/* CT24 Chuong trinh tim so duong trong day so */
#include <stdio.h>
main()
{
int i,n,t;
float x[100];

printf(" VAO SO LIEU \n");
printf("n = ");scanf("%d",&n);
for (i=0;i<n;i++)
    {
    printf("x[%d] = ",i);
```

```
    scanf("%f",&x[i]);
    }
t=0;
for (i=0;i<n;i++)
    if (x[i]>0)
        {
        printf("x[%d] = %10.2f\n",i,x[i]);
        t++;
        }
printf("So duong = %d\n",t);
getch();
return;
}
```

Ta được kết quả chương trình CT24 như sau :

VAO SO LIEU

n = 5

x[0] = 23

x[1] = -15.68

x[2] = 38

x[3] = 490

x[4] = 57.65

x[0] =   23.00

x[2] =   38.00

x[3] =   490.00

x[4] =   57.65

So duong = 4

Bài số 25 Tính trung bình số học của dãy n số thực trong một khoảng [a, b].

Bài giải Để tính trung bình số học của một dãy n số thực trong một khoảng [a, b] ta thực hiện một chu trình để kiểm tra các số x[i] có nằm trong khoảng [a, b], nếu đúng sẽ thực hiện tính tổng theo thuật toán đã trình bày trong bài số 23.

```
/* CT25 Chuong trinh tinh trung binh trong day so */
#include <stdio.h>
main()
{
int i,n,t;
float a,b,m,x[100];

printf("\n Chuong trinh tinh trung binh\n");
printf(" VAO SO LIEU\n");
printf(" a = ");scanf("%f",&a);
printf(" b = ");scanf("%f",&b);
printf(" n = ");scanf("%d",&n);
for (i=0;i<n;i++)
    {
    printf("x[%d] = ",i);scanf("%f",&x[i]);
    }
m=0;
t=0;
for (i=0;i<n;i++)
    {
    if (x[i]>a)
        if (x[i]<b)
```

```
        {
        t++;
        m+=x[i];
        }
    }
m=m/t;
printf(" TRUNG BINH TRONG KHOANG [a,b] = %f\n",m);
getch();
return;
}
```

Trong chương trình nếu cho các cận a,b và dãy n số thực như sau sẽ có kết quả :

Chuong trinh tinh trung binh

a = 2

b = 10

n = 4

x[0] = 3.5

x[1] = 4.75

x[2] = -12.45

x[3] = 4.75

TRUNG BINH TRONG KHOANG [a,b] = 4.333333

Bài số 26 Tính 100 số đầu tiên của các số Fibonacci.

Bài giải Số Fibonacci được xác định theo công thức truy hồi sau :

$u_n = u_{n-1} + u_{n-2}$, $n = 3,4$; $u_1 = 1$, $u_2 = 2$

Ta thực hiện một chu trình for để tính lặp và ghi các số Fibonacci trong số n cho trước

```
/* CT26 Chuong trinh tinh chuoi FIBONACCI */
#include <stdio.h>
main()
{
int n,i,u[100];

printf("\n Chuong trinh tinh cac so FIBONACCI\n");
printf(" VAO SO LIEU \n");
printf(" n = ");scanf("%d",&n);
u[1]=1;
u[2]=2;
for (i=3;i<=n;i++)
    {
    u[i]=u[i-1]+u[i-2];
    printf("%d ",u[i]);
    }
getch();
return;
}
```

Kết quả với n = 10 ta có các số Fibonacci như sau :

Chuong trinh tinh cac so FIBONACCI

VAO SO LIEU

n = 10

3 5 8 13 21 34 55 89

Bài số 27 Xác định số lớn nhất và số các số dương trong dãy n số thực.

Bài giải Gọi M = max x[i] và t là số các số dương trong dãy số đã cho. Đầu tiên coi M = x_1 và kiểm tra điều kiện $x_i > 0$ để đếm các số dương, sau đó so sánh M với các giá trị x_i để tìm giá trị lớn nhất. Thuật toán kết thúc khi đã kiểm tra hết n số. In giá trị lớn nhất M và số các số dương t.

```
/* CT27 Chuong trinh xac dinh max trong day so */
#include <stdio.h>
main()
{
int t,i,n;
float max,x[100];

printf(" Chuong trinh xac dinh so lon nhat\n");
printf(" VAO SO LIEU\n");
printf(" n = ");scanf("%d",&n);
for (i=0;i<n;i++)
    {
    printf("x[%d] = ",i);
    scanf("%f",&x[i]);
    }
max = x[0];
t =. 0;
for (i=0;i<n;i++)
    {
    if (x[i]>0) t = t+1;
        {
```

```
        if (max>x[i]) max = max;
        else max = x[i];
    }
}
printf(" MAX = %f trong do co %d so la so duong\n",max,t);
getch();
return;
}
```

Trong chương trình nếu vào các dữ liệu sau đây sẽ có kết quả :

Chuong trinh xac dinh so lon nhat

VAO SO LIEU

n = 4

x[0] = 10

x[1] = -25.57

x[2] = 120

x[3] = 23.65

MAX = 120.000000 trong do co 3 so la so duong

Bài số 28 Tính các phần tử của tam giác Pascal.

Bài giải Tam giác pascal được xác định như sau :

C_0^0

$C_1^0\ C_1^1$

$C_2^0\ C_2^1\ C_2^2$

. . . .

$C_n^0\ C_n^1\ C_n^2 \ldots C_n^n$

trong đó C_n^k là tổ hợp chập k trong n phần tử được xác định theo công thức truy hồi :

$$C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$$

Cột đầu có tất cả các phần tử đều bằng 1, đồng thời giả thiết

$C_n^n = 1$.

Trong chương trình có hàm NhiThuc (int n, int p) để thực hiện tính tổ hợp C theo công thức truy hồi.

```
/* CT28A Chuong trinh tinh tam giac PASCAL */
#include <stdio.h>
int NhiThuc(int,int); /* prototype */

main()
{
int n,m,p;

printf("\n Chuong trinh tinh tam giac Pascal\n");
printf(" Cho so m = ");scanf("%d",&m);
for (n=1;n<m+1;n++)
   {
   for (p=1;p<n+1;p++)
      {
      printf("%d ",NhiThuc(n,p));
      }
   printf("\n"); /* De xuong mot dong */
}
getch();
return;
}
```

```
/* Ham Nhi thuc */
int NhiThuc(int n, int p)
{
if (p==1 || p==n) return(1);
else return(NhiThuc(n-1,p-1)+NhiThuc(n-1,p));
}
```

Kết quả nếu cho các dữ liệu vào như sau với chương trình CT28A:

Chuong trinh tinh tam giac Pascal

Cho so m = 10

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

1 6 15 20 15 6 1

1 7 21 35 35 21 7 1

1 8 28 56 70 56 28 8 1

1 9 36 84 126 126 84 36 9 1

```
/* CT28B Chuong trinh tinh tam giac PASCAL */

#include <stdio.h>
#include <conio.h>
#define max 15
```

```
main()
{
int t[max], /* Bang mot dong cua tam giac */
    n,          /* So dong thuc te */
    i,j         /* Chi so dong, cot */
clrscr();
printf("Cho so dong n=");scanf("%d",&n);
if(n>max) n=max;
printf("\n\n p ");
for(i=0;i<n;i++)
    printf("%5d",i);
    printf("\n n\n");
for(i=0;i<=n;i++)
    printf("*****");
    printf("\n");
for(i=0;i<n;i++)
    {
    t[i]=1;
    for(j=i-1;j>0;j--)
        t[j]=t[j-1]+t[j];
        printf("%2d--",i);
    for(j=0;j<=i;j++)
        printf("%5d",t[j]);
        printf("\n");
    }
getch();
return;
}
```

Kết quả của chương trình CT28B :

Cho so dong n = 10

p / n	0	1	2	3	4	5	6	7	8	9
***										
0 --	1									
1 --	1	1								
2 --	1	2	1							
3 --	1	3	3	1						
4 --	1	4	6	4	1					
5 --	1	5	10	10	5	1				
6 --	1	6	15	20	15	6	1			
7 --	1	7	21	35	35	21	7	1		
8 --	1	8	28	56	70	56	28	8	1	
9 --	1	9	36	84	126	126	84	36	9	1

Bài số 29 Đảo ngược một dãy n số thực.

Bài giải Thuật toán gồm việc đọc dãy n số thực a[i], i = 1...n và đảo ngược chỉ số của dãy i = n...1 và in kết quả của dãy.

```
/* CT29 Chuong trinh dao nguoc mot day so */
#include <stdio.h>
main()
{
int i,jn;
float tg,a[100];
```

```
printf("\n Chuong trinh dao nguoc mot day so\n");
printf(" Vao so con so cua day so n = ");
scanf("%d",&n);
for (i=1;i<=n;i++)
    {
    printf(" a[%d] = ",i);
    scanf("%f",&a[i]);
    }
printf("\n Day so dau tien\n");
for (i=1;i<=n;i++)
    printf("%5.2f ",a[i]);
for (i=1,j=n;i<j;++i,--j)
    {
    tg = a[i];
    a[i] = a[j];
    a[j] = tg;
    }
printf("\n Day so dao nguoc\n");
for (i=1;i<=n;++i)
    printf(" %5.2f ",a[i]);
getch();
return;
}
```

Trong chương trình nếu cho n dãy số a_i ta sẽ được kết quả như sau

Chuong trinh dao nguoc mot day so

Vao so con so cua day so n = 4

a[1] = 12

a[2] = 35

a[3] = 46

a[4] = 78

Day so dau tien

12.00   35.00   46.00   78.00

Day so dao nguoc

78.00   46.00   35.00   12.00

Bài số 30 Sắp xếp dãy n số thực theo thứ tự tăng hoặc giảm

Bài giải Thuật toán sắp xếp một dãy số như sau :

- Đọc mảng n số thực.

- Để sắp xếp ta lấy phần tử đầu tiên và lần lượt so sánh với các phần tử tiếp theo nếu lớn hơn thì đổi chỗ cho giá trị của phần tử so sánh, tiếp tục kiểm tra và đổi chỗ cho đến khi đã kiểm tra hết n số thực và ghi kết quả dãy số đã sắp xếp.

```
/* CT30 Chuong trinh sap xep day so theo chieu giam */
#include <stdio.h>
main()
{
int i,j,n;
float t,M[100];
printf("\n Chuong trinh sap xep mot mang n so\n");
printf("Cho n = ");scanf("%d",&n);
for (i=0;i<n;i++)
    {
    printf("M[%d] = ",i);
    scanf("%f",&M[i]);
    }
```

```
for (i=0;i<n;i++)
    for (j=0; j<n; j++)
        {
        if (M[i]>M[j])
            {
            t=M[i];
            M[i]=M[j];
            M[j]=t;
            }
        }
printf("\n Ket qua sau khi sap xep\n");
for (i=0;i<n;i++)
    printf("\nM[%d] = %f ",i,M[i]);
getch();
return;
}
```

Trong chương trình nếu lấy n = 5 và cho dãy số m_i như sau sẽ có kết quả :

Chuong trinh sap xep motmang n so

Cho n = 4

M[0] = 12

M[1] = 45

M[2] = 67

M[3] = 89

Ket qua sau khi sap xep

M[0] = 89.000000

M[1] = 67.000000

M[2] = 45.000000

M[3] = 12.000000

Bài số 31 Giải phương trình trùng phương $ax^4 + bx^2 + c = 0$

Bài giải Phương trình trùng phương được giải theo thuật toán sau:

delta = b^2 - 4ac

Nếu delta > 0 có 2 nghiệm: t_1 = (-b + sqrt(delta)/ (2*a)

t_2 = (-b - sqrt(delta)/ (2*a)

nếu $t_1 \geq 0$ có nghiệm thứ nhất $x_1 = \sqrt{t_1}$

có nghiệm thứ hai $x_2 = -\sqrt{t_1}$

nếu $t_2 \geq 0$ nghiệm thứ ba $x_3 = \sqrt{t_2}$

nghiệm thứ tư $x_4 = -\sqrt{t_2}$

nếu không sẽ vô nghiệm.

Nếu delta = 0 có nghiệm kép t = - b/2a

nghiệm thứ nhất $x_1 = \sqrt{t_1}$

nghiệm thứ hai $x_2 = -\sqrt{t_1}$

Nếu delta < 0 phương trình vô nghiệm.

```
/* CT31 Chuong trinh giai phuong trinh trung phuong */
#include <stdio.h>
#include <math.h>

void main ()
{
float a,b,c,delta,t1,t2,tg;
printf("\n Giai phuong trinh trung phuong\n");
printf("\n Vao he so a = "); scanf("%f",&a);
```

```
printf("\n Vao he so b = "); scanf("%f",&b);
printf("\n Vao he so c = "); scanf("%f",&c);
delta=b*b-4*a*c;
if (delta>=0)
{
if (delta>0)     /* Delta duong */
    {
    t1=(-b+sqrt(delta))/(2*a);
    t2=(-b-sqrt(delta))/(2*a);
    if (t1>=0)
        {
        tg=sqrt(t1);
        printf(" Nghiem thu nhat x1  = %8.3f\n",tg);
        printf(" Nghiem thu hai x2  = %8.3f\n",-tg);
        }
    if (t2>=0)
        {
        tg=sqrt(t2);
        printf(" Nghiem thu ba x3  = %8.3f\n",tg);
        printf(" Nghiem thu tu x4  = %8.3f\n",-tg);
        }
    if ((t1<0)&&(t2<0))
        printf("Phuong trinh vo nghiem\n ");
    }  /* delta>0 */
else if (delta==0)
    {
    t1 = -b/(2*a);
        if (t1>=0)
```

```
            {
            tg=sqrt(t1);
            printf (" Nghiem thu nhat x1 = %8.3f\n", tg);
            printf (" Nghiem thu nhat x2 = %8.3f\n",-tg);
            }
        else
            printf(" Phuong trinh vo nghiem\n");
        } /* delta = 0*/
    }
else if(delta<0)
printf("Phuong trinh vo nghiem\n");
getch ();
return;
}
```

Trong chương trình 31 nếu cho các dữ liệu như sau :

Giai phuong trinh trung phuong

Vao he so a = 2.5

Vao he so b = -8.65

Vao he so c = 1.25

Nghiem thu nhat x1 = 1.819

Nghiem thu hai x2 = -1.891

Nghiem thu ba x3 = 0.389

Nghiem thu tu x4 = - 0.389

Bài số 32 Gọi a, b, c là ba cạnh của một tam giac là các số dương bat kỳ. Nếu ba số đó có thể là các canh của tam giác hãy tính diện tích của

tam giác đó. Thử điều kiện đó là tam giác thường hay tam giác vuông, tam giác cân hoặc tam giác đều.

Bài giải Thuật toán như sau :

- Đọc ba số dương a,b,c bất kỳ, nếu chúng là các cạnh của một tam giác thì chiều dài của một cạnh phải nhỏ hơn tổng chiều dài của hai cạnh kia. Nếu đúng sẽ tính nửa chu vi của tam giác theo công thức :

$$p = \frac{a + b + c}{2}$$

và tính diện tích tam giác theo công thức Heron :

$$S = \sqrt{p(p-a)(p-b)(p-c)}$$

- Tam giác là vuông nếu thoả mãn định lý Pitago $c^2 = a^2 + b^2$

- Tam giác là đều khi a = b = c, là cân khi a = b, hoặc b = c, hoặc c = a

- Trong chương trình nếu không thỏa mãn điều kiện các cạnh của một tam giác sẽ ghi dòng chữ KHONG TAO THANH TAM GIAC

```
/* CT32 Chuong trinh tinh tam giac */
#include <stdio.h>
#include <math.h>

main()
{
float a,b,c,p,s;

printf("\n\n\t\tChuong trinh tinh tam giac\n");
printf("Vao a,b,c la cac canh cua tam giac\n");
printf("a = ");scanf("%f",&a);
printf("b = ");scanf("%f",&b);
printf("c = ");scanf("%f",&c);
```

```
if((a>0)&&(b>0)&&(c>0)&&(a<(b+c))&&(b<(a+c))&&(c<(a+b)))
   {
   printf("\n Ba so da cho la ba canh cua tam giac\n");
   if ((a==b)||(b==c)||(a==c))
      if ((a==b)&&(b==c))
      printf("\n Day la tam giac deu\n");
   else printf("\n Day la tam giac can\n");
   if ((a*a+b*b==c*c)||(b*b+c*c==a*a)||(a*a+c*c==b*b))
      printf("\n Day la tam giac vuong\n");
   p=(a+b+c)/2;
   s=sqrt(p*(p-a)*(p-b)*(p-c));
   printf(" Chu vi tam giac la : %8.2f\n",2*p);
   printf(" Dien tich tam giac la : %8.2f\n",s);
   }
else
printf(" KHONG TAO THANH TAM GIAC\n");
getch();
return;
}
```

Nếu cho các cạnh như sau sẽ có kết quả :

Chuong trinh tam giac

Vao a,b,c la cac canh cua tam giac

a = 4

b = 3

c = 5

Ba so da cho la ba canh cua tam giac

Day la tam giac vuong

Chu vi tam giac la : 12.00

Dien tich tam giac là : 6.00

Bài số 33 Tính giai thừa n! của một số nguyên

Bài giải Theo công thức định nghĩa của giai thừa

$$n! = n(n - 1)(n - 2)...1$$

ta sẽ lập một chu trình tính tích theo công thức truy hồi

p = p.k với k là số nguyên, khi k > n sẽ ra khỏi chu trình và ghi kết quả p.

Ta cũng có thể tính giai thừa theo công thức đệ quy bằng cách định nghĩa hàm GTDe quy (n) và hàm Giai Thua (n) theo công thức truy hồi.

Chương trình tổ chức tính giai thừa theo công thức lặp và theo giai thừa đệ quy.

```
/* CT33 Chuong trinh tinh giai thua */
#include <stdio.h>
long GiaiThua(int);
long GTDequy(int);

main()
{
int n;
tt: printf("\n Cho mot so nho hon 16 n = ");scanf("%d",&n);
printf("\n Giai thua cua %d la : %20ld\n",n,GiaiThua(n));
printf("\n Giai thua de quy cua %d :%20ld\n",n,GTDequy(n));
printf(" Tiep tuc hoac stop (go 1 hoac 0) :");scanf("%d",&n);
printf("\n");
```

```
if (n==1) goto tt;
getch();
return;
}
long GiaiThua(n)
int n;
{
long s=1,i;
if (n==1 || n==0)
    return(1);
else
    {
    for (i=1;i<=n;++i) s *=i;
    return(s);
    }
}

long GTDequy(n)
int n;
{
if (n==1||n==0) return(1);
else return(n*GTDequy(n-1));
}
```

Nếu cho n ta được kết quả như sau :

Cho mot so nho hon 16 n = 10

Giai thua cua 10 la : 3628800

Giai thua de quy cua 10 : 3628800

Tiep tuc hoac stop (go 1 hoac 0) : 0

Bài số 34 Xác định các số nguyên tố trong khoảng từ 1 đến n với n là số tự nhiên cho trước.

Bài giải Có nhiều thuật toán khác nhau để tìm các số nguyên tố trong khoảng 1 đến n.

Thuật toán của Eratosthène như sau :

Từ tập các số nguyên từ 1 đến n mỗi lần gặp một số nguyên tố sẽ bị loại trừ ra khỏi tập này tất cả các số là bội của số nguyên tố này. Eratosthène là nhà toán học từ thời chưa có giấy viết. Vì vậy bài toán tìm số nguyên tố được ông đặt ra và thực hiện trên mặt đất bằng cách kẻ một dãy ô trong đó ghi các số tự nhiên từ 1 đến n. Số nào không phải là số nguyên tố thì bị loại ra bằng cách chọc một lỗ vào ô đó để xóa đi. Kết quả trên mặt đất lỗ chỗ như một cái sàng lọc các số, vì thế bài toán có tên là *Sàng Eratosthène*.

Chương trình CT34A xác định các số nguyên tố từ 1 đến n.

```
/* CT34A Sang Eratosthene dynamic */
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#define true 1
#define false 0

main()
{
unsigned n,     /* So luong cac so nguyen */
    *Gachdi,    /* con tro cac so khong ngto */
    soNgto,
    i;
int na;         /* Bo dem so cua sang */
```

```
/* Cho n va dinh vi mang */
clrscr();
do
    {
    printf("\nCho so luong cac so can xet n = ");
    scanf("%u",&n);
    Gachdi = (unsigned *)malloc((n+1)*sizeof(unsigned));
    if (Gachdi == NULL)
        printf("\nKhong du bo nho\n");
    }
while (Gachdi == NULL);
/* Khoi tao sang */
for (i=1;i<=n;i++) /* Gan cac so 0 cho sang */
    Gachdi[i] = false;
Gachdi[1] = true;   /* Gach so 1 vi khong la so nguyen to */
/* Chuyen vao sang */
soNgto = 1;
while (soNgto*soNgto<=n)
    {
    while (Gachdi[++soNgto] && soNgto<n) {}
    /* Tim so dau tien khong bi Gach di */
    for (i=2*soNgto;i<=n;i+=soNgto) /* Gach cac boi so cua so
ngto */
    Gachdi[i] = true;
    }
/* Hien thi ket qua */
printf("\nGiua 1 va %d co cac so nguyen to la : \n\n",n);
na = 0;
```

```
for (i=1;i<=n;i++)
if ( !Gachdi[i]) /* Khong nam trong mang gach di - do la so
ngto */
{
printf("%6u",i);
na++ ;
if(na%150==0) /* Mot trang man hinh chua 15 dong = 150 so
ng to */
{
printf("\n"); /* Xuong dong sau khi het mot trang */
printf("An mot phim bat ky de xem tiep\n");
getch();
}
else
if (na%10==0) printf("\n"); /* 10 so mot dong */
}
printf("\nTrong khoang tu 1 den %d co %d so nguyen to.",n,na);
getch();
return;
}
```

Nếu cho n ta được kết quả như sau :

Cho so luong cac so can xet n = 25

Giua 1 va 25 co cac so nguyen to la :

2 3 5 7 11 13 17 19 23

Trong khoang tu 1 den 25 co 9 so nguyen to.

Chương trình CT34B xác định các số nguyên tố trong một khoảng [n1,n2]

```
/* CT34B Chuong trinh Sang Eratosthene */
#include <stdio.h>
#include<conio.h>
main()
{
int i,j,n1,n2,na;
clrscr();
printf("\nChuong trinh tim so nguyen to giua khoang 2 so\n");
printf("\nVao so tu nhien thu nhat n1 = ");
scanf("%d",&n1);
printf("\nVao so tu nhien thu hai n2 (n2<=32767) = ");
scanf("%d",&n2);
na = 0;
for (i=n1;i<=n2;i++)
   {
   for (j=2;j<=i-1;j++)
      if (i%j==0) goto aaa;
   printf("%d\t",i);
   na++;
   if (na%150==0)      /* Mot trang man hinh chua 150 so
nguyen to */
      {
      printf("\n"); /* Xuong dong sau khi het mot trang */
      printf("An mot phim bat ky de xem tiep\n");
      getch();
      }
```

```
    aaa:;
    }
getch();
return;
}
```

Kết quả chương trình sẽ là :

Chuong trinh tim so nguyen to giua khoang 2 so

Vao so tu nhien thu nhat n1 = 10

Vao so tu nhien thu hai n2 (n2 < = 32767) = 25

11 13 17 19 23

Bài số 35 Lập chương trình thực hiện các phép tính cộng, trừ, nhân, chia, khai căn bậc hai và lũy thừa bậc n của số phức

Bài giải Cũng như ngôn ngữ PASCAL, ở ngôn ngữ C không có định nghĩa số phức. Tuy vậy chúng ta cũng có thể thực hiện các phép tính đối với các số phức bằng cách dùng các cấu trúc có dạng :

```
struct {
    float re;
    float im;
}sophuc;
```

Ta xây dựng các hàm thực hiện các phép tính đối với số phức theo thuật toán sau :

Cộng trừ $(x_1 \pm iy_1) + (x_2 \pm iy_2) = (x_1 \pm x_2) + i(y_1 \pm y_2)$

Nhân $(x_1 \pm iy_1)(x_2 \pm iy_2) = (x_1x_2 - y_1y_2) + i (x_1y_2 + x_2y_1)$

Chia $\dfrac{x_1 + iy_1}{x_2 + iy_2} = \dfrac{x_1x_2 + y_1y_2}{x_2^2 + y_2^2} + \dfrac{i(y_1x_2 - x_1y_2)}{x_2^2 + y_2^2}$

Căn bậc hai $x + iy = (a + ib)^{1/2}$, trong đó

$$x = [\ a/2 + ((a/2)^2 + (b/2)^2)^{1/2}\ ]^{1/2}$$

$$y = [\ -a/2 + ((a/2)^2 + (b/2)^2)^{1/2}\ ]^{1/2}$$

nếu $a > 0$ ta tính x và $y = b/2x$

nếu $a < 0$ ta tính y và $x = b/2y$

Luỹ thừa $(a + ib)^n = r^n.e^{in\theta} = r^n(\cos n\theta + i \sin n\theta)$

Trong đó $r = (a^2 + b^2)^{1/2}$

$$\theta = \mathrm{arctg}\,\frac{b}{a}$$

Trong chương trình CT35 gồm tổ chức thực hiện 5 hàm sau :

- Tổng hai số phức
- Tích hai số phức
- Thương hai số phức
- Căn bậc hai của một số phức
- Luỹ thừa bậc n của một số phức

Chương trình chính gồm việc đưa các số phức cần tính , gọi 5 hàm và ghi kết quả.

```
/* CT35 Chuong trinh tinh toan so phuc */
#include <stdio.h>
#include <math.h>
typedef struct {        /* cau truc so phuc */
          float re;
          float im;
          } sophuc;
void TONG(sophuc,sophuc,sophuc); /* prototype ham phuc */
void HIEU(sophuc,sophuc,sophuc);
void TICH(sophuc,sophuc,sophuc);
```

```
void THUONG(sophuc,sophuc,sophuc);
void CAN(sophuc,sophuc);
void LuyThua(sophuc,sophuc);

main ()
{
sophuc x,y,z;
printf(" Vao so phuc x\n");
printf("x.re = ");
scanf("%f",&x.re);
printf("x.im = ");
scanf("%f",&x.im);
printf(" Vao so phuc y\n");
printf("y.re = ");
scanf("%f",&y.re);
printf("y.im = ");
scanf("%f",&y.im);
z.re=0;
z.im=0;
TONG(x,y,z);
TICH(x,y,z);
THUONG(x,y,z);
CAN(x,z);
LuyThua(x,z);
getch();
return;
}
```

```
void TONG(sophuc a,sophuc b,sophuc c)
{
c.re=a.re+b.re;
c.im=a.im+b.im ;
printf(" Tong la (%6.1f) + j(%5.1f)\n",c.re,c.im);
return;
}

void HIEU(sophuc a,sophuc b,sophuc c)
{
c.re=a.re-b.re;
c.im=a.im-b.im;
printf(" Hieu la (%6.1f) + j(%5.1f)\n",c.re,c.im);
return;
}

void TICH(sophuc a,sophuc b,sophuc c)
{
c.re=a.re*b.re-a.im*b.im;
c.im=a.re*b.im+a.im*b.re;
printf(" Tich la (%6.1f) + j(%5.1f)\n",c.re,c.im);
return;
}

void THUONG(sophuc a,sophuc b,sophuc c)
{
c.re=(a.re*b.re+a.im*b.im)/(b.re*b.re+b.im*b.im);
c.im=(a.im*b.re-a.re*b.im)/(b.re*b.re+b.im*b.im);
printf(" Thuong la (%6.1f) + j(%5.1f)\n",c.re,c.im);
```

```
return;
}

void CAN(sophuc a,sophuc c)
{
printf("Khai can mot so phuc : \n");
printf("Cho so phuc : \n");
printf("co phan thuc : ");
scanf("%f",&a.re);
printf("co phan ao : ");
scanf("%f",&a.im);
if (a.re>0)
    {
    c.re=sqrt(a.re/2+sqrt(((a.re/2)*(a.re/2))+((a.im/2)*(a.im/2))));
    c.im=a.im/(2*c.re);
    }
else
    {
    c.im=sqrt(-a.re/2+sqrt(((a.re/2)*(a.re/2))+((a.im/2)*(a.im/2))));
    c.re=a.im/(2*c.im);
    }
printf(" Can la (%6.1f) + j(%5.1f)\n",c.re,c.im);
return;
}

void LuyThua(sophuc a,sophuc c)
{
int n;
float r,t;
```

```
printf("Luy thua mot so phuc : \n");
printf("Cho so phuc : \n");
printf("co phan thuc : ");
scanf("%f",&a.re);
printf("co phan ao : ");
scanf("%f",&a.im);
printf("Cho bac cua luy thua n = ");
scanf("%d",&n);
r = sqrt((a.re*a.re)+(a.im*a.im));
t = atan(a.im/a.re);
c.re = pow(r,n)*cos(n*t);
c.im = pow(r,n)*sin(n*t);
printf("Luy thua bac %d cua so phuc (%f)+j(%f) la\n",n,a.re,a.im);
printf(" (%f) +j(%f)\n",c.re,c.im);
return;
}
```

Vao so phuc x

x.re = 5

x.im = -7.5

Vao so phuc y

y.re = 6.75

y.im = 1.25

Tong la (11.8) + j (-6.2)

Tich la (43.1) + j (-44.4)

Thuong la (0.5) +j (-1.2)

Khai can mot so phuc ;

Cho so phuc :

co phan thuc : 5.75

co phan ao : -12.25

Can la (3.1) +j (-2.0)

Luy thua mot so phuc :

Cho so phuc :

co phan thuc : 8.25

co phan ao : -7.15

Cho bac cua luy thua n = 3

Luy thua bac 3 cua so phuc (8.250000) + j(-7.150000) la

(-703.766357) + j (-1094.414795)

Bài số 36 Giải phương trình bậc hai với a,b,c là các số phức dạng

$$ax^2 + bx + c = 0$$

Bài giải Ta áp dụng thuật toán giải phương trình bậc hai quen thuộc với điều kiện các hệ số a,b,c và nghiệm được biểu diễn dưới dạng cấu trúc có hai thành phần thực và ảo như trong bài số 35

```
/* CT36 Giai phuong trinh bac hai he so phuc */
#include <stdio.h>
#include <math.h>
typedef struct {
            float real;
            float imag;
            } sophuc;

main ()
{
sophuc a,b,c,s1,s2,u,v,w;
double r;
```

```
printf("Vao so phuc a\n");
printf("a.real = ");
scanf("%f",&a.real);
printf("a.imag = ");
scanf("%f",&a.imag);
printf("Vao so phuc b\n");
printf("b.real = ");
scanf("%f",&b.real);
printf("b.imag = ");
scanf("%f",&b.imag);
printf("Vao so phuc c\n");
printf("c.real = ") ;
scanf("%f",&c.real);
printf("c.imag = ");
scanf("%f",&c.imag);
u.real=b.real*b.real-b.imag*b.imag-4*a.real*c.real+4*a.imag*c.imag;
u.imag = 2*b.real*b.imag-4*a.real*c.imag-4*a.imag*c.real;
r = sqrt(u.real*u.real+u.imag*u.imag);
v.real = sqrt((r+u.real)/2);
v.imag = (u.imag<0)?-sqrt((r-u.real)/2) :sqrt((r-u.real)/2);
w.real = (-b.real - v.real) / 2;
w.imag = (-b.imag - v.imag) / 2;
u.real = (-b.real + v.real) / 2;
u.imag = (-b.imag + v.imag) / 2;
r = a.real*a.real + a.imag*a.imag;
s1.real = (a.real*w.real + a.imag*w.imag) / r;
s1.imag = (a.real*w.imag - a.imag*w.real) / r;
s2.real = (a.real*u.real + a.imag*u.imag) / r;
```

```
s2.imag = (a.real*u.imag - a.imag*u.real) / r;
printf("Phuong trinh co hai nghiem la\n");
printf("s1 = (%6.2f) + j(%6.2f)\n",s1.real,s1.imag);
printf("s2 = (%6.2f) + j(%6.2f)\n",s2.real,s2.imag);
getch();
return;
}
```

Kết quả nếu ta đưa các dữ liệu vào CT36 sẽ có kết quả như sau :

Vao so phuc a
a.real = 1.75
a.imag = 3.25
Vao so phuc b
b.real = -4.12
b.imag = 5.75
Vao so phuc c
c.real = 1.25
c.imag = 2.15
Phuong trinh co hai nghiem la
s1 = (-0.10) + j(0.30)
s2 = (-0.74) + j(-2.02)

Bài số 37 Tính tổng và tích của hai số phức dùng con trỏ

Bài giải Cũng như các bài toán đối với số phức, ta cần định nghĩa một cấu trúc kiểu

```
struct
    { double real;
    double imag;
    } sophuc;
```

Để thuận tiện ta xây dựng hai hàm tính tổng và tích có biến là các con trỏ dạng :

```
void TinhTong ( sophuc * , sophuc * , sophuc * )
void TinhTich ( sophuc * , sophuc * , sophuc * )
```

```
/* CT37 Chuong trinh tong va tich hai phuc so dung pointer */
#include<stdio.h>
typedef struct
        { double real;
        double imag;
        } sophuc;
void TinhTong(sophuc *,sophuc *,sophuc *);
void TinhTich(sophuc *,sophuc *,sophuc *);

main()
{
sophuc z1,z2,tong,tich;
z1.real = 0.5; z1.imag = 1.0;
z2.real = 1.0; z2.imag = 1.0;
TinhTong(&z1,&z2,&tong);
TinhTich(&z1,&z2,&tich);
printf("Hai so phuc (%4.2f)+j(%4.2f) va (%4.2f)+j(%4.2f)\n",
    z1.real,z1.imag,z2.real,z2.imag);
printf("co tong la (%4.2f)+j(%4.2f) \n",tong.real,tong.imag);
printf("co tich la (%4.2f)+j(%4.2f) \n\n",tich.real,tich.imag);
```

```
printf(" an Enter de tiep tuc \n");
getch(); /* Dung man hinh de xem ket qua */
return; /* Phai co cau lenh nay neu dung TC version 2.0 */
}

void TinhTong(sophuc *x,sophuc *y,sophuc *tong)
{
tong->real = x->real + y->real;
tong->imag = x->imag + y->imag;
}

void TinhTich(sophuc *x,sophuc *y,sophuc *tich)
{
tich->real = x->real * y->real - x->imag * y->imag;
tich->imag = x->real * y->imag + x->imag * y->real;
}
```

Các dữ liệu đưa vào trong chương trình và cho ta kết quả như sau:

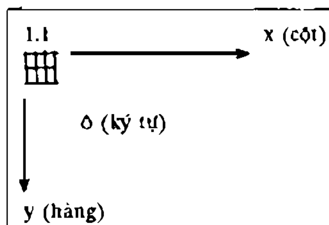
Hai so phuc (0.50) + j(1.00) va (1.00) + j(1.00)
co tong la (1.50) + j(2.00)
co tich la (-0.50) + j(1.50)
an Enter de tiep tuc

Chương 3

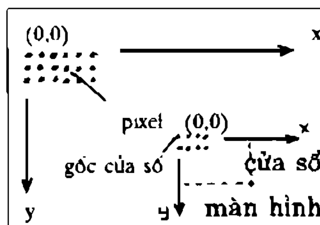
ĐỒ HỌA

Khả năng đồ họa phong phú là một trong những ưu điểm nổi bật của ngôn ngữ C. Chương này sẽ giới thiệu một số chương trình vẽ các hàm giải tích trong mặt phẳng và trong không gian.

Trước hết ta cần phân biệt màn hình ở chế độ đồ họa. Ở chế độ văn bản màn hình được tạo bởi các ô thường có 25 hàng và 80 cột. Mỗi ô biểu diễn cho một ký tự. Góc trái phía trên ứng với ô (1,1) và số cột theo chiều từ trái sang phải, số hàng theo chiều từ trên xuống dưới (hình 3-1).



Chế độ văn bản



Chế độ đồ họa

Hình 3-1

Ở chế độ đồ hoạ, màn hình gồm ma trận các điểm sáng (pixel). Mỗi điểm có khả năng tạo nên một hay nhiều màu. Tùy theo chế độ đồ họa và loại màn hình sử dụng chiều rộng và chiều cao của ma trận điểm sáng có thể là 640x350 (EGA 16 Color mode), 640x480 (VGA 16 Color mode). Gốc toạ độ ở góc trên phía trái, chiều ox từ trái sang phải, chiều oy từ trên xuống dưới. Cửa sổ màn hình được xác định bằng setviewport là hình chữ nhật chứa mọi đường vẽ. Các đường nằm ngoài cửa sổ này sẽ bị xén mất. Trên hình 3-1 vẽ cửa sổ màn hình ở giữa màn

Để xây dựng một chương trình đồ hoạ đầu tiên ta phải khởi tạo hệ thống đồ họa, xác định loại màn hình, mốt đồ họa sẽ sử dụng. Tiếp theo cần xác định màu nền, màu đường vẽ, màu tô, kiểu tô, cuối cùng là biểu diễn các hàm giải tích theo các tọa độ sao cho có thể sử dụng được các hàm của chế độ đồ họa. Khởi tạo chế độ đồ họa bằng

initgraph (&graphdriver, &graphmode, " ");

trong đó graphdriver là một biến nguyên chứa giá trị số của thiết bị màn hình hoặc chứa hằng DETECT, graphmode là một biến nguyên. Khi graphdriver chứa giá trị số của thiết bị màn hình thì graphmode chứa một trong các mốt đồ họa tương ứng với màn hình này. Khi nội dung của graphdriver bằng DETECT thì không cần khởi đầu cho graphmode, " " là tên của thư mục chứa bộ chương trình đồ họa.

Để chọn màu nền ta dùng hàm setbkcolor (color). Để chọn màu đường vẽ ta dùng hàm setcolor (color). Để chọn kiểu tô và màu tô ta dùng hàm setfillstyle (pattern, color). Trong cả ba trường hợp color là biến nguyên chứa mã của màu. Biến pattern chứa mã của màu tô. Trong phần phụ lục có giới thiệu một số hàm đồ họa thông dụng.

Bài số 38 Vẽ đường biểu diễn hàm y = f(x)

Ứng dụng vẽ cho $y_1 = \sin x$

$$y_2 = \frac{\sin x}{x}$$

Bài giải Đây là hàm một biến cho dưới dạng tường

Để khởi tạo chế độ đồ họa ta dùng hàm intKhoiTaoDoHoa (void) trong đó có dùng câu lệnh if để xác định loại màn hình sử dụng có phải thuộc loại VGA hoặc CGA không. Dùng setbkcolor (CYAN) để đặt màu nền và mở cửa sổ màn hình bằng setviewport (0, getmaxy/2, getmaxx(), getmaxy(), 0); cho phép vẽ hình cả ngoài cửa sổ. Vẽ trục x, trục y, viết tên trục x và y cũng như gốc (0,0)

Dùng settextstyle để định kiểu chữ cho dòng DO THI Y = f(X)

Để vẽ đồ thị hàm y = f(x) cần phải cho x biến thiên một lượng Δx và tính giá trị hàm y tương ứng. Để làm điều này chỉ cần một chu trình for để biến đổi x,y. Kết quả ta được một dãy trị số x,y. Để vẽ từng điểm ta dùng hàm putpixel (x,y,color), ở đây x,y là cặp giá trị x,y nguyên và color là màu đường muốn vẽ.

```
/* CT38 Chuong trinh ve ham y = f(x) */

#include <stdlib.h>
#include <math.h>
#include <graphics.h>
#define pi 3.14159255
#define TileX 20
#define TileY 100
int i,x,y;
float t;
int KhoiTaoDoHoa(void);

main()
{
KhoiTaoDoHoa();
setbkcolor(CYAN);
```

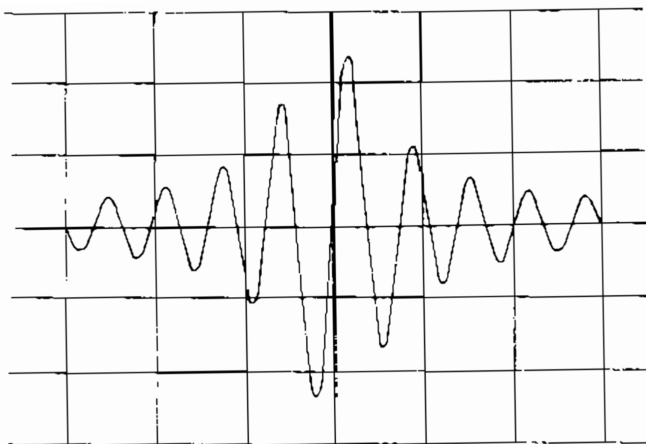
```
/* Mo cua so va cho phep ve hinh ca ngoai cua so */
setviewport(0,getmaxy()/2,getmaxx(),getmaxy(),0);
setcolor(LIGHTRED);
outtextxy(getmaxx()/2,10,"(0,0)");
setcolor(RED);
moveto(getmaxx()/2-5,10);      /* Chuyen con tro ve goc va ve
truc Y */
linerel(0,-(getmaxy()/2-20));
outtextxy((getmaxx()/2-5),-(getmaxy()/2-20),"y");
moveto(getmaxx()/2-5,10);      /* Chuyen con tro ve goc va ve
truc Y */
linerel(0,getmaxy()/2-60);
moveto(0,0);        /* ve truc X */
linerel(getmaxx()-10,0);
outtextxy(getmaxx()-10,0,"x");
setcolor(BLUE);
settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
outtextxy(200,200,"DO THI Y = F(X)");
for (i=1;i<=2000;i++)      /* Ham y = sin(x) */
    {
    x = (int)(2*pi*i*TileX/400);
    y = - (int)(sin((2*i*pi)/400)*TileY);
    putpixel(x,y,YELLOW);
    }
getch();
for (i=1;i<=2000;i++)      /* Ham y =sin(x)/x */
    {
    t = pi*i;
```

```
    x = (int) (t*TileX/400+320);
    y = - (int)(2*sin((t)/400)*TileY/t*200);
    putpixel(x,y,WHITE);
    }
getch();
closegraph();
return;
}

int KhoiTaoDoHoa(void)
{
int graphdriver = DETECT, graphmode, error_code;
initgraph(&graphdriver, &graphmode, "..\\bgi");
error_code = graphresult();
if (error_code!= grOk)
return(-1);  /* Khong phai man EGA hoac VGA */
if ((graphdriver!= EGA) && (graphdriver!= VGA))
    {
    closegraph();
    return 0;
    }
return(1);
}
```

Để vẽ hàm y = sinx ta chọn tỷ lệ xích của x là 20, của y là 100.

Hàm $y = \frac{\sin x}{x}$ với $x \in [0,20]$ có dạng biểu diễn trên hình 3-2.



Hình 3-2 $y = \frac{\sin x}{x}$

Bài số 39 Vẽ đường biểu diễn hàm xác định theo tham số

$$\begin{cases} x = f(t) \\ y = g(t) \end{cases}$$

Ứng dụng vẽ đồ thị hàm hypocycloide $x = \cos^3(t)$

$y = \sin^3(t)$, $t \in [\ 0{,}2\pi\ ]$

hàm trochoide $x = 2t - 3\sin(t)$

$y = 2 - 3\cos(t)$, $t \in [\ -9{,}9\ ]$ và

hàm $x = (1+\cos^2(t)).\sin(t)$

$y = \sin^2(t).\cos(t)$, $t \in [\ 0{,}2\pi\ ]$

Bài giải Các đường cong theo tham số được xác định bằng hai phương trình. Bằng cách biến thiên tham số t ta được dãy các cặp (x,y) tương ứng rồi dùng hàm putpixel (x,y, color) ; để vẽ liên tiếp các điểm. Ta nhận thấy hàm y = f(x) là một trường hợp riêng của các hàm thông số khi lấy x = t ta được :

$$\begin{cases} x = t \\ y = g(t) = g(x) \end{cases}$$

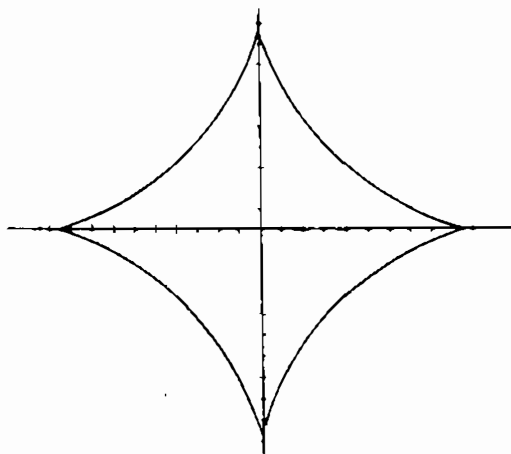
```
/* CT39 Chuong trinh ve duong cong theo tham so */
#include <stdlib.h>
#include <math.h>
#include <graphics.h>
int i,x,y;
int KhoiTaoDoHoa(void);

main()
{
float t;
KhoiTaoDoHoa();
setbkcolor(CYAN);
/* Mo cua so va cho phep ve hinh ca ngoai cua so */
setviewport(0,getmaxy()/2,getmaxx(),getmaxy(),0);
setcolor(LIGHTRED);
outtextxy(getmaxx()/2,10,"(0,0)");
setcolor(RED);
moveto(getmaxx()/2-5,10); /* Chuyen con tro ve goc va ve truc Y */
linerel(0,-(getmaxy()/2-20));
outtextxy((getmaxx()/2-5),-(getmaxy()/2-20),"y");
moveto(getmaxx()/2-5,10); /* Chuyen con tro ve goc va ve truc Y */
linerel(0,getmaxy()/2-60);
moveto(0,0);    /* ve truc X */
linerel(getmaxx()-10,0);
outtextxy(getmaxx()-10,0,"x");
setcolor(BLUE);
settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
outtextxy(200,200,"DO THI HAM THEO THAM SO");
```

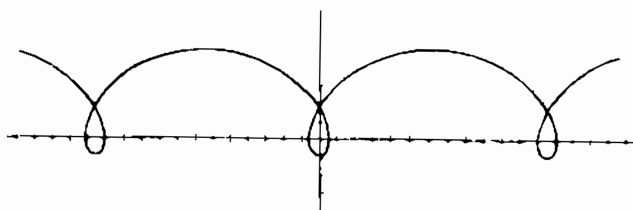
```
for (i=0;i<=1000;i++)      /* Duong Hypocycloide */
    {
    t = 0.00628*i;
    x = (int)(cos(t)*cos(t)*cos(t)*120+320);
    y = (int)(sin(t)*sin(t)*sin(t)*120);
    putpixel(x,y,YELLOW);
    }
getch();
for (i=-1000;i<=1000;i++)      /* Duong Trocoide */
    {
    t= 0.01*i;
    x = (int)((2*t-3*sin(t))*15+320);
    y = -(int) ((2-3*cos(t))*10);
    putpixel(x,y,WHITE);
    }
getch();
for(i=0;i<=1000;i++)      /* Duong so 3 */
    {
    t= 0.00628*i;
    x = (int)(((1+cos(t)*cos(t))*sin(t))*150+320);
    y = (int)(sin(t)*sin(t)*cos(t)*200);
    putpixel(x,y,RED);
    }
getch();
closegraph();
return;
}
```

```
int KhoiTaoDoHoa(void)
{
int graphdriver = DETECT, graphmode, error_code;
initgraph(&graphdriver, &graphmode, "..\\bgi");
error_code = graphresult();
if (error_code!= grOk)
return(-1);       /* Khong phai man EGA hoac VGA */
if ((graphdriver!= EGA) && (graphdriver!= VGA))
    {
    closegraph();
    return 0;
    }
return(1);
}
```

Kết quả ta được các đường biểu diễn sau đây (hình 3-3.a, b và c).

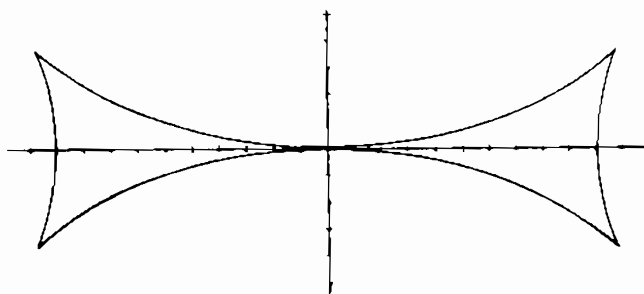


Hình 3-3 a Đường hypocycloide $x = \cos^3(t)$
$y = \sin^3(t)$



Hình 3-3 b Đường trochoide $x = 2t - 3\sin(t)$

$y = 2 - 3\cos(t)$



Hình 3-3 c Đường $x = (1+\cos2(t))\sin(t)$

$y = \sin2(t).\cos(t$

Bài số 40 Biểu diễn đường cong theo tọa độ cực $r = f(\theta)$

Ứng dụng vẽ đường xoắn ốc Archimede $r = \theta$, $\theta \in [0,40]$ và đường hoa hồng bốn cánh $r = \sin2\theta$, $\theta \in [0,2\pi]$

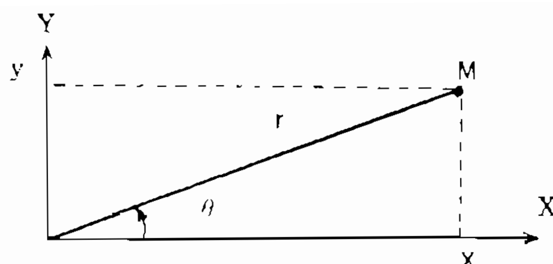
Tọa độ cực r,θ và tọa độ xoy được biểu diễn trên hình 3-4.

Ta có quan hệ $x = r\cos\theta$

$y = r\sin\theta$

và trở về việc vẽ đồ thị hàm số theo tham số. Ta dùng chu trình for để biến đổi góc θ kết quả thu được các cặp x,y ứng với mỗi giá trị của θ.

Đồ thị thu được vẽ trên hình 3-5 a và b



Hình 3-4

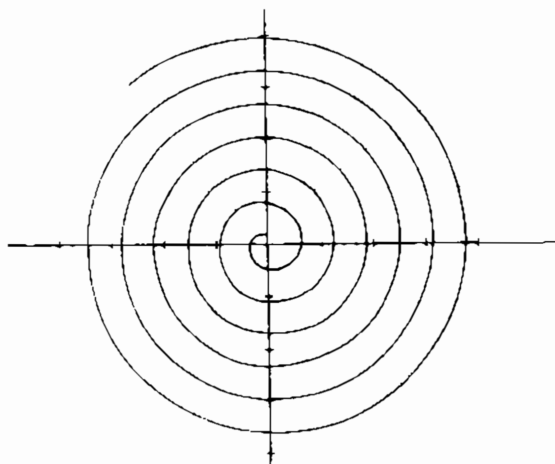
```
/* CT40 Chuong trinh do hoa theo toa do cuc */

#include <stdlib.h>
#include <math.h>
#include <graphics.h>
int i,x,y;
int KhoiTaoDoHoa(void);

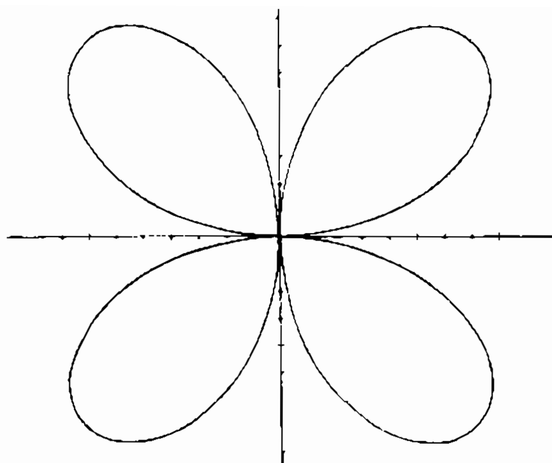
main()
{
float r, teta;
KhoiTaoDoHoa();
setbkcolor(WHITE);
/* Mo cua so va cho phep ve hinh ca ngoai cua so */
setviewport(0,getmaxy()/2,getmaxx(),getmaxy(),0);
setcolor(LIGHTRED);
outtextxy(getmaxx()/2,10,"(0.0)");
setcolor(RED);
moveto(getmaxx()/2-5,10); /* Chuyen con tro ve goc va ve truc Y */
linerel(0,-(getmaxy()/2-20));
```

```
outtextxy((getmaxx()/2-5),-(getmaxy()/2-20),"y");
moveto(getmaxx()/2-5,10); /* Chuyen con tro ve goc va ve truc Y */
linerel(0,getmaxy()/2-60);
moveto(0,0).            /* ve truc X */
linerel(getmaxx()-10,0);
outtextxy(getmaxx()-10,0,"x");
setcolor(BLUE);
settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
outtextxy(200,200,"DUONG CONG THEO TOA DO CUC ");
for (i=0;i<=2000;i++)          /* Duong xoan oc Archimede */
    {
    teta = 0.02*i;
    x = (int)(teta*cos(teta)*3+320);
    y = -(int)(teta*sin(teta)*3);
    putpixel(x,y,CYAN);
    }
getch();
for(i=0;i<=2000;i++)        /* Duong hoa hong 4 canh */
    {
    teta=0.00628*i;
    x = (int)(sin(2*teta)*cos(teta)*150+320);
    y = (int)(sin(2*teta)*sin(teta)*150);
    putpixel(x,y,RED);
    }
getch();
closegraph();
return;
}
```

```
int KhoiTaoDoHoa(void)
{
int graphdriver = DETECT, graphmode, error_code;
initgraph(&graphdriver, &graphmode, " \\bgi");
error_code = graphresult();
if (error_code!= grOk)
return(-1);      /* Khong phai man EGA hoac VGA */
if ((graphdriver!= EGA) && (graphdriver!= VGA))
   {
   closegraph();
   return 0;
   }
return(1);
}
```



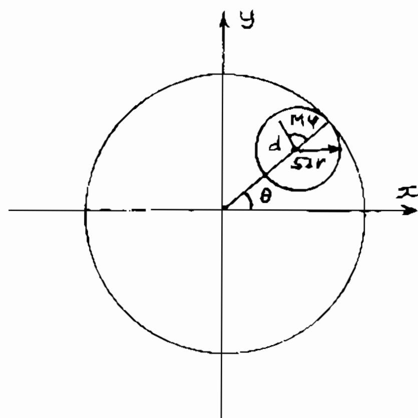
Hình 3-5 a $r=\theta$ đường xoắn ốc Archimede



Hình 3-5 b $r = \sin 2\theta$, đường hoa hồng bốn cánh

Bài số 41 Vẽ đường cong cycloide

Một đĩa tâm Ω bán kính r quay bên trong và tiếp xúc với vòng tròn tâm O bán kính R.



Hình 3-6

Điểm M trên đĩa mô tả đường cong cycloide (hình 3-6)

Đường cycloide là trường hợp riêng của đường cong theo tham số. Gọi M là vị trí của điểm cần tìm quỹ đạo, O là tâm đường tròn, Ω là tâm của đĩa. Gọi d = ΩM là khoảng cách giữa M và tâm đĩa và đặt $d = k\frac{r}{10}$. Như vậy khi k biến thiên giữa 0 và 10 điểm M chạy từ tâm vòng tròn đến chu vi. Khi k càng gần 10 đường cong càng nhọn ngược lại k càng gần 0

đường cong càng tròn. Ta có :

$$OM = O\Omega + \Omega M$$

$$x = (R - r)\cos\theta + d\cos(\theta + \varphi)$$

$$y = (R - r)\sin\theta + d\sin(\theta + \varphi)$$

Vì đĩa quay trong vòng tròn ta có $r\varphi = -R\theta$. Khử φ ta được

$$x = (R - r)\cos\theta + \frac{kr}{10}\cos[(1 - \frac{R}{r})\theta]$$

$$y = (R - r)\sin\theta + \frac{kr}{10}\sin[(1 - \frac{R}{r})\theta]$$

Chọn R = 1, đặt $r = \frac{n}{N}$ ta có :

$$x = (1 - \frac{n}{N})\cos\theta + k\frac{n}{10N}\cos[(1 - \frac{N}{n})\theta$$

$$y = (1 - \frac{n}{N})\sin\theta + k\frac{n}{10N}\sin[(1 - \frac{N}{n})\theta$$

Đây là phương trình tham số theo θ. Cho θ biến thiên từ O đến 2π, phụ thuộc vào trị số của n, N và k ta được các quỹ đạo cycloide khác nhau (hình 3-7)

```
/* CT41 Chuong trinh ve quy dao Cycloide */
#include <stdlib.h>
#include <math.h>
#include <graphics.h>
int i,x,y;
int KhoiTaoDoHoa(void);

main()
{
int n,N,k;
float teta;
printf("\n Vao du lieu ");
```

```
printf("\n Cho n = "); scanf("%d",&n);
printf("\n Cho N = "); scanf("%d",&N);
printf("\n Cho k = "); scanf("%d",&k);
KhoiTaoDoHoa();
setbkcolor(CYAN);
/* Mo cua so va cho phep ve hinh ca ngoai cua so */
setviewport(0, getmaxy()/2, getmaxx(),getmaxy(),0);
setcolor(LIGHTRED);
outtextxy(getmaxx()/2,10,"(0.0)");
setcolor(RED);
moveto(getmaxx()/2-5,10); /* Chuyen con tro ve goc va ve truc Y */
linerel(0,-(getmaxy()/2-20));
outtextxy((getmaxx()/2-5),-(getmaxy()2-20),"y");
moveto(getmaxx()/2-5),10); /* Chuyen con tro ve goc va ve truc Y */
linerel(0,getmaxy()/2-60);
moveto(00);                /* ve truc X */
linerel(getmaxx()-10,0);
outtextxy(getmaxx()-10,0"x");
setcolor(BLUE);
settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
outtextxy(200,200,"DO THI QUY DAO");
for (i= 0;i<=1000;i+ +)
      {
      teta = 0.00628*i;
      x =(int)(((1-n/N)*cos(teta)+k*n*cos((1-N/n)*teta)/(10*N))
         *150 + 320);
      y = (int)(((1-n/N)*sin(teta)+k*n*sin((1-N/n)*teta)/(10*N))*150);
```

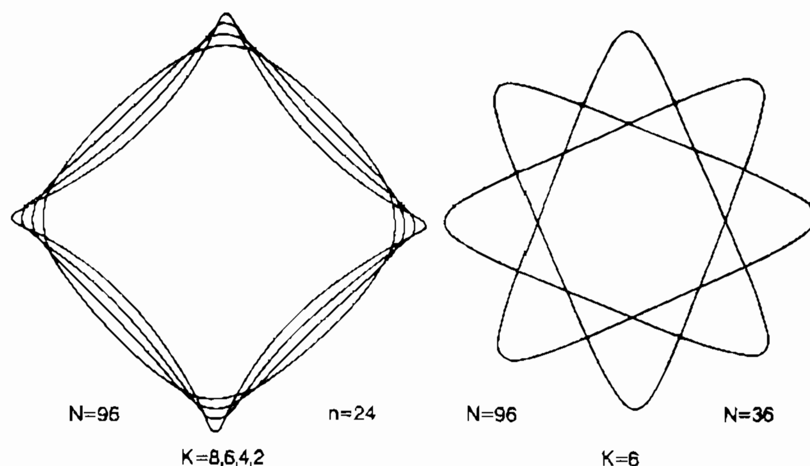
```
    putpixel(x,y,YELLOW);
    }
getch();
closegraph();
return;
}

int KhoiTaoDoHoa(void)
{
int graphdriver = DETECT, graphmode, error_code;
initgraph(&graphdriver, &graphmode, "..\\bgi");
error_code = graphresult();
if (error_code!= grOk)
return(-1);       /* Khong phai man EGA hoac VGA */
if ((graphdriver!= EGA) && (graphdriver!= VGA))
    {
    closegraph();
    return 0;
    }
return(1);
}
```

Bài số 42 Vẽ mặt trong không gian ba chiều hàm hai biến z = f(x,y).

Trường hợp cụ thể z = 5sinxsiny

Bài giải Ta cần tìm cách biểu diễn không gian lên mặt phẳng. Vì màn hình chỉ có hai chiều, để vẽ hình không gian ba chiều ta phải chiếu không gian lên mặt phẳng. Có nhiều phương pháp chiếu, ở đây ta sử dụng phép chiếu song song. Màn hình có tọa độ OXY và tọa độ không gian oxyz (hình 3-8) Oz là trục thẳng đứng, góc giữa oX và ox là α góc



Hình 3-7

oX và oy là β. Điểm O (X_o,Y_o) có toạ độ màn hình là :

$$Oo = X_o\vec{I} + Y_o\vec{J}$$

$$\vec{i} = \cos(\alpha)\vec{I} + \sin(\alpha)\vec{J}$$

$$\vec{j} = \cos(\beta)\vec{I} + \sin(\beta)\vec{J}$$

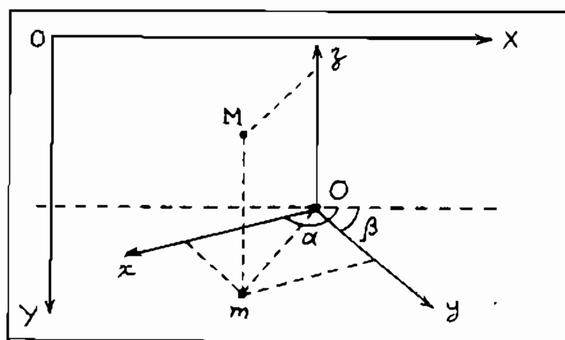
$$\vec{k} = -\vec{J}$$

Gọi M là một điểm trong không gian các toạ độ là các số thực (x,y,z).

Chiếu lên mặt phẳng màn hình có tọa độ (X,Y)

$$oM = x\vec{i} + y\vec{j} + z\vec{k}$$

$$OM = X\vec{I} + Y\vec{J}$$



Hình 3-8

Ta được các công thức chuyển từ tọa độ không gian về tọa độ màn hình :

$$X = Xo + x\cos(\alpha) + y\cos(\beta)$$

$$Y = Yo + x\sin(\alpha) + y\sin(\beta) - z$$

Trong hệ thống ta xác định một điểm trong không gian bằng 2 số thực và 5 số nguyên :

- hai số nguyên cho tọa độ gốc O trong OXY

- ba số nguyên tính bằng pixel cho các đơn vị trên trục ox,oy,oz

- hai số thực chỉ góc giữa OX và các trục ox,oy. Trong chương trình tọa độ màn hình được định nghĩa bằng cấu trúc gr3_toado.

Để chuyển toạ độ cần xây dựng hai hàm :

int X (double x, double y, double z)

int Y (double x, double y, double z) theo các công thức chuyển tọa độ.

Hàm cần vẽ được định nghĩa bằng hàm hai biến dạng :

double ham (double x, double y);

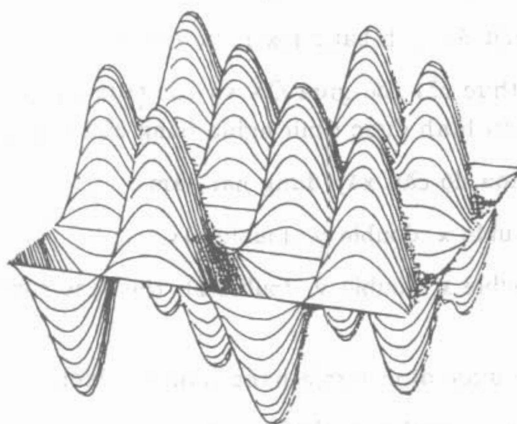
Mặt được biểu diễn bằng dãy liên tiếp các đường cong cắt bằng các mặt phẳng song song theo mặt xoy hoặc yoz tùy theo vị trí của người quan sát. Mỗi đường cong được tạo nên bởi dãy các đoạn thẳng nối với các điểm tính toán trên mặt. Đa số mặt với độ chính xác chấp nhận cần vẽ khoảng 50 đường cong và số điểm tính toán khoảng 100.

Để mặt biểu diễn dễ quan sát nên loại bỏ các đường khuất theo nguyên lý sau đây :

Theo đường vẽ ta xác định hai đường biên biểu diễn các điểm cực đại và cực tiểu do các đường cong khác nhau tạo ra trên mặt. Khi vẽ đường mới vị trí của các điểm của đoạn đang vẽ được so sánh với vùng khuất bao gồm giữa hai đường biên.

Trong chương trình phần khởi tạo chế độ đồ họa, xây dựng cấu trúc toạ độ màn hình struct gr3_toado, xây dựng hàm chuyển tọa độ X và Y, xây dựng hàm z = ham(x,y) tương tự bài trên.

Để vẽ có loại bỏ đường khuất ta tạo nên một mảng gọi tên là mat[8] để chứa tọa độ X,Y để kiểm tra điều kiện tô màu và dùng fillpoly (4,mat) ; để vẽ và tô màu đa giác có đỉnh là các điểm (x1,y1)...(x4,y4). Kết quả ta được hình biểu diễn của hàm z = sinxsiny (hình 3-9).



Hình 3-9

```
/* CT42 Chuong trinh ve mat khong gian 3 chieu mat an */
#include <stdio.h>
#include <math.h>
#include <graphics.h>
typedef struct              /* toa do man hinh */
   { int OX,        /* truc hoanh man hinh */
   OY,              /* truc tung man hinh */
   UX,              /* don vi truc hoanh */
```

```
    UY,       /* don vi truc tung */
    UZ;       /* don vi truc oz */
double Xx,  /* goc (OX,ox) */
    Xy ;      /* goc (OX,oy) */
    } gr3_toado;
gr3_toado gr3 ={ 250,20,20,20,20,0.8*M_PI,0.2*M_PI};
int gr3_cla = GREEN,      /* mau cac truc */
    gr3_clt = YELLOW,     /* mau duong */
    gr3_clf = BLUE,     /* mau nen */
    gr3_clc = RED;      /* mat an */
double gr3_lx = 2,
    gr3_ly = 2,
    gr3_lz = 2;
int X(double x,double y,double z);   /* chuyen toa do */
int Y(double x,double y,double z);
double ham (double,double);          /* ham can ve */
int KhoiTaoDoHoa(void);

main()
{
double x,y,z, xa=-6.28,xb=6.28,ya=-6.28,yb=6.28,xp=0.2,yp=0.2;
int mat[8];
KhoiTaoDoHoa();
setbkcolor(CYAN);
/* Mo cua so va cho phep ve hinh ca ngoai cua so */
setviewport(0,getmaxy()/2,getmaxx(),getmaxy(),0);
setcolor(LIGHTRED);
outtextxy(getmaxx()/2,10,"(0,0)");
```

```
setcolor(RED);
moveto(getmaxx()/2-5,10); /* Chuyen con tro ve goc va ve truc Y */
linerel(0,-(getmaxy()/2-20));
outtextxy((getmaxx()/2-5),-(getmaxy()/2-20),"y");
moveto(getmaxx()/2-5,10); /* Chuyen con tro ve goc va ve truc Y */
linerel(0,getmaxy()/2-60);
moveto(0,0);        /* ve truc X */
linerel(getmaxx()-10,0);
outtextxy(getmaxx()-10,0,"x");
setcolor(BLUE);
settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
outtextxy(200,-220,"DO THI KHONG GIAN 3 CHIEU");
for (x=xa;x<=xb;x+=xp)  /* Ve co the hien mat an */
   for(y=ya ;y<=yb ;y+=yp)
      {
      if( kbhit() ) return;
      z= ham(x,y);              /* diem A */
      mat[0] = X(x,y,z);
      mat[1] = Y(x,y,z);
      z =ham(x,y+yp);           /* diem B */
      mat[2] = X(x,y+yp,z);
      mat[3] = Y(x,y+yp,z);
      z = ham(x+xp,y+yp);   /* diem C */
      mat[4] = X(x+xp,y+yp,z);
      mat[5] = Y(x+xp,y+yp,z);
      z = ham(x+xp,y);          /* diem D */
      mat[6] = X(x+xp,y,z) ;
      mat[7] = X(x+xp,y,z) ;
```

```
        if ( (mat[3]-mat[1])*(mat[6]-mat[0])-
            (mat[7]-mat[1])*(mat[2]-mat[0])<0)
            setfillstyle(1,YELLOW);
        else
            setfillstyle(1,GREEN);
        fillpoly(4,mat);
    }
    closegraph();
    return;
}

int X(double x,double y,double z)     /* Ham chuyen toa do */
{
return( (gr3.OX+
    x*gr3.UX*cos(gr3.Xx)+
    y*gr3.UY*cos(gr3.Xy)));
}

int Y(double x,double y,double z)
{
return( (gr3.OY+
    x*gr3.UX*sin(gr3.Xx)+
    y*gr3.UY*sin(gr3.Xy)-z*gr3.UZ));
}

double ham (double x, double y)     /* Ham can ve */
{
return( 5*sin(x)*sin(y));
}
```

```
int KhoiTaoDoHoa(void) /* Ham khoi tao che do do hoa */
{
int graphdriver = DETECT, graphmode, error_code;
initgraph(&graphdriver, &graphmode, "..\\bgi");
error_code = graphresult();
if (error_code!= grOk)
    return(-1); /* Khong phai man EGA hoac VGA */
if ((graphdriver!= EGA) && (graphdriver!= VGA))
    {
    closegraph();
    return 0;
    }
return(1);
}
```

Bài số 43 Vẽ mặt trong không gian ba chiều **hàm hai biến** $z = f(x,y)$

Trường hợp cụ thể $z = \dfrac{5\sin(x^2 + y^2)}{x^2 + y^2}$

Bài giải Tương tự bài số 42 ở đây ta cũng sử dụng phép chiếu không gian lên mặt phẳng và dùng các công thức chuyển đổi từ toạ độ không gian về tọa độ màn hình.

Chương trình CT43 gồm các bước sau :

- Xác định màn hình bằng cấu trúc gr3_toado
- Xây dựng hai hàm chuyển toạ độ :

```
int X (double x,double y, double z)
int Y (double x,double y, double z)
```

- Xác định hàm cần vẽ đồ thị bằng :

    double ham (double x, double y) ;

- Cho x biến thiên từ -10 đến +10 với bước $x_p = 0,1$

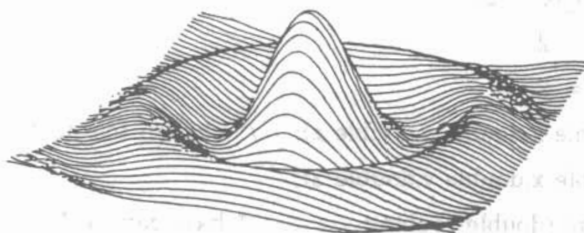
  Cho y biến thiên từ -10 đến +10 với bước $y_p = 0,1$

gọi hàm z = ham (x,y);

và dùng lineto (X(x,y,z), Y(x,y,z));

để vẽ đường thẳng từ điểm hiện tại tới điểm (X,Y) và chuyển con chạy tới vị trí điểm (X,Y). Kết quả ta được mặt biểu diễn hàm hai biến

$z = \frac{5 \sin \sqrt{x^2 + y^2}}{\sqrt{x^2 + y^2}}$ bằng phép chiếu song song (hình 3-10)



Hình 3-10

```
/* CT43 Chuong trinh ve mat khong gian 3 chieu */
#include <stdio.h>
#include <math.h>
#include <graphics.h>
```

```
typedef struct              /* toa do man hinh */
  { int OX,          /* truc hoanh man hinh */
     OY,             /* truc tung man hinh */
     UX,            /* don vi truc hoanh */
     UY,            /* don vi truc tung */
     UZ;            /* don vi truc oz */
double Xx,          /* goc (OX,ox) */
  Xy;               /* goc (OX,oy) */
  } gr3_toado;
gr3_toado gr3 ={ 320,25,20,20,20,0.8*M_PI,0.2*M_PI};
  int gr3_cla = GREEN,      /* mau cac truc *
  gr3_clt = YELLOW,         /* mau duong */
  gr3_clf = BLUE,           /* mau nen */
  gr3_clc = RED;            /* mat an */
double gr3_lx = 2,
  gr3_ly = 2,
  gr3_lz = 2;
int X(double x,double y,double z);    /* chuyen toa do */
int Y(double x,double y,double z);
double ham (double,double);           /* ham can ve */
int KhoiTaoDoHoa(void);

main()
{
double x,y,z, xa=-10,xb=10,ya=-10,yb=10,xp=0.1,yp=0.1;
int mat[8];
KhoiTaoDoHoa();
setbkcolor(CYAN);
```

```
/* Mo cua so va cho phep ve hinh ca ngoai cua so */
setviewport(0,getmaxy()/2,getmaxx(),getmaxy(),0);
setcolor(LIGHTRED);
outtextxy(getmaxx()/2,10,"(0,0)");
setcolor(RED) ;
moveto(getmaxx()/2-5,10); /* Chuyen con tro ve goc va ve truc Y */
linerel(0,-(getmaxy()/2-20));
outtextxy((getmaxx()/2-5),(gemaxy()/2-20),"y");
moveto(getmaxx()/2-5,10); /* Chuyen con tro ve goc va ve truc Y */
linerel(0,getmaxy()/2-60);
moveto(0,0);                    /* ve truc X */
linerel(getmaxx()-10,0);
outtextxy(getmaxx()-10,0,"x");
setcolor(YELLOW);
settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
outtextxy(200,-240,"DO THI KHONG GIAN 3 CHIEU");
for (x=xa;x<=xb;x+=xp)  /* Ve do thi */
      for(y=ya;y<=yb;y+=yp)
           { if( kbhit() ) return;
           z= ham(x,y);
           lineto(X(x,y,z), Y(x,y,z));
           }
      closegraph();
      return;
}
int X(double x,double y,double z)          /* Ham doi toa do */
{ return( (gr3.OX+
```

```
    x*gr3.UX*cos(gr3.Xx)+
    y*gr3.UY*cos(gr3.Xy)));
}

int Y(double x,double y,double z)
{ return( (gr3.OY+
    x*gr3.UX*sin(gr3.Xx)+
    y*gr3.UY*sin(gr3.Xy)-z*gr3.UZ));
}

double ham (double x, double y)     /* Ham can ve do thi */
{
return( 5*sin(sqrt(x*x+y*y))/sqrt(x*x+y*y));
}

int KhoiTaoDoHoa(void)     /* Ham khoi tao che do do hoa */
{
int graphdriver = DETECT, graphmode, error_code;
initgraph(&graphdriver, &graphmode, "..\\bgi");
error_code = graphresult();
if (error_code!= grOk)
    return(-1);          /* Khong phai man EGA hoac VGA */
if ((graphdriver!= EGA) && (graphdriver!= VGA))
    {
    closegraph();
    return 0;
    }
return(1);
}
```

Bài số 44 Vẽ biểu đồ thống kê các dữ liệu.

Bài giải Chương trình chính gồm việc nhập các dữ liệu, tính giá trị phần trăm của dữ liệu và in kết quả theo giá trị và theo phần trăm. Tiếp theo là vẽ biểu đồ dạng hình hộp.

Dùng hàm bar3d để vẽ các khối hộp chữ nhật có kích thước chiều cao phụ thuộc vào giá trị thống kê. Số lượng khối hình chữ nhật bằng số giá trị thống kê, ở đây số giá trị thống kê cực đại chọn là 6. Kết quả cho trên hình 3-11.

```
/* CT44 Chuong trinh ve bieu do thong ke */
#include <stdio.h>
#include <stdlib.h>
#include <float.h>
#include <graphics.h>
#include <math.h>
#include <conio.h>
#define MAX 100
#define SoGiaTri 6
void VeBieuDo(float p[]);        /* Ham ve bieu do */
void main(void)
{
int i;
int giatri[SoGiaTri];
float phantram[SoGiaTri];
for (i = 0; i < SoGiaTri; i++)
    {
    printf("\nCho gia tri thong ke (giua 0 va %d) : ",MAX);
    scanf("%d", &giatri[i]);
    }
```

```
for (i = 0; i < SoGiaTri; i++)
    phantram[i] = ((float) giatri[i]) / MAX;
printf("\n\n\nNo\tGIA TRI\t\tSo %");
for (i = 0;i<SoGiaTri;i++)
    printf("\n%d\t%d\t\t%3.0f",i+1,giatri[i],(phantram[i]*100));
getch();
VeBieuDo(phantram);
}
void VeBieuDo(float p[])
{
int gd, gm;
int i, trai, tren, rong, duoi, sau;
detectgraph(&gd,&gm);
initgraph(&gd,&gm,"");
setbkcolor(GREEN);
setcolor(RED);
settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
outtextxy(200,30,"BIEU DO THONG KE") ;
rong = (int)((getmaxx()) / ((SoGiaTri * 2 ) + 1));
duoi = getmaxy() - 20;
sau = (int) (rong / 4);
trai = rong;
for (i = 0; i < SoGiaTri; i++)
    {
    tren = (duoi) - ((int)(p[i] * 300));
    bar3d(trai,tren,(trai + rong),duoi,sau,1);
    trai += (rong * 2);
```

```
getch();
closegraph();
return;
}
```

Cho gia tri thong ke (giua 0 va 100) : 48

Cho gia tri thong ke (giua 0 va 100) : 35

Cho gia tri thong ke (giua 0 va 100) : 29

Cho gia tri thong ke (giua 0 va 100) : 71

Cho gia tri thong ke (giua 0 va 100) : 15

Cho gia tri thong ke (giua 0 va 100) : 38

No	*GIA TRI*	*So%*
1.	*46*	*46*
2.	*35*	*35*
3.	*29*	*29*
4.	*71*	*71*
5.	*15*	*15*
6.	*38*	*38*



Hình 3-11

Bài số 45 Vẽ hệ mặt trời

Bài giải Hệ mặt trời gồm các hành tinh, tính từ vị trí gần mặt trời nhất là :

Mercury (sao Thủy) Venus (sao Kim) Earth (Trái đất) Mars (sao Hỏa) Jupiter (sao Mộc) Saturn (sao Thổ) Uranus (sao Hải vương).

Chương trình chính ngoài hàm khởi tạo chế độ đồ họa còn có :

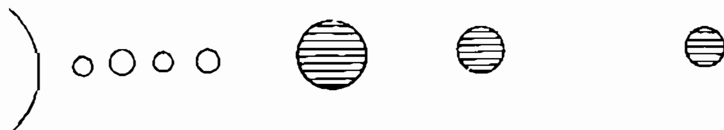
- Hàm tính tọa độ hành tinh void TinhToaDo(void), xác định tọa độ của các hành tinh dùng các biến nguyên au1 là đơn vị thiên văn tính bằng pixel cho hành tinh nhỏ hơn trái đất.

au2- đơn vị thiên văn tính bằng pixel cho các hành tinh lớn hơn trái đất.

erad - bán kính trái đất tính bằng pixel.

- Hàm vẽ hành tinh , vẽ vòng tròn có tọa độ và bán kính xác định theo tỷ lệ so với bán kính trái đất. Mặt trời được biểu diễn bằng một cung tròn.

Hình vẽ hệ mặt trời cho trên hình 3-12.



Hình 3-12

```
/* CT45 Chuong trinh ve he Mat troi */
#include <graphics.h>
#include <stdlib.h> /* Dung cho lenh exit() */
```

```
#include <stdio.h>
#include <conio.h>
int KhoiTaoDoHoa(void);
void TinhToaDo(void);  /* Ty le khoang cach tren man hinh */
void VeCacHanhTinh(void);
void VeHanhTinh(float ToaDo_x, float Bankinh,
        int Mau, int Kieu_to);
void get_key(void);        /* wait for key */
/* Bien tong the cho ham TinhToaDo */
int max_x, max_y;          /* Gia tri Max cua truc x va truc y */
int y_org;                 /* Truc y cho toan hinh ve */
int au1;                   /* Don vi thien van tinh bang pixels
                           cho hanh tinh nho hon Trai dat */
int au2;                   /* Don vi thien van tinh bang pixels
                           cho hanh tinh lon hon Trai dat */
int erad;                  /* Ban kinh Trai dat bang pixels */

int main()
{
/* Kiem tra xem man hinh la EGA hoac VGA */
if (KhoiTaoDoHoa() != 1)
   {
   printf("\nChuong trinh nay dung cho man hinh EGA
        hoac VGA\n");
   exit(0);
   }
TinhToaDo();      /* Thang ti le theo do phan giai man hinh */
VeCacHanhTinh();
get_key();
```

```
closegraph();
return 0;
}

int KhoiTaoDoHoa(void)
{
int graphdriver = DETECT, graphmode, error_code;
initgraph(&graphdriver, &graphmode, "..\\bgi");
error_code = graphresult();
if (error_code!= grOk)
   return(-1);      /* Khong phai man EGA hoac VGA */
if ((graphdriver!= EGA) && (graphdriver!= VGA))
   {
   closegraph();
   return 0;
   }
return(1);
}

void TinhToaDo(void)
{
max_x = getmaxx();
max_y = getmaxy();
y_org = max_y / 2;      /* Toa do Y cho moi vat the */
erad = max_x / 200;     /* Ban kinh Trai dat tinh bang pixels */
au1 = erad * 20;        /* Ti le cho hanh tinh nho hon Trai dat */
au2 = erad * 10;        /* Ti le cho hanh tinh lon hon Trai dat */
}
```

```
void VeCacHanhTinh()
{
/* Cung Mat troi */
VeHanhTinh(-90, 100, YELLOW, EMPTY_FILL);

/* Mercury */
VeHanhTinh(0.4 * au1, 0.4 * erad, BROWN, LTBKSLASH_FILL);

/* Venus */
VeHanhTinh(0.7 * au1, 1.0 * erad, WHITE, SOLID_FILL);

/* Earth */
VeHanhTinh(1.0 * au1, 1.0 * erad, LIGHTBLUE, SOLID_FILL);

/* Mars */
VeHanhTinh(1.5 * au1, 0.4 * erad, LIGHTRED,
CLOSE_DOT_FILL);

/* Jupiter */
VeHanhTinh(5.2 * au2, 11.2 * erad, WHITE, LINE_FILL);

/* Saturn */
VeHanhTinh(9.5 * au2, 9.4 * erad, LIGHTGREEN, LINE_FILL);

/* Uranus */
VeHanhTinh(19.2 * au2, 4.2 * erad, GREEN, LINE_FILL);
}

void VeHanhTinh(float ToaDo_x, float Bankinh, int Mau, int
      Kieu_to)
{
setcolor(Mau);
```

```
circle(ToaDo_x, y_org, Bankinh);
setfillstyle(Kieu_to, Mau);
floodfill(ToaDo_x, y_org, Mau);
}

void get_key(void)
{
outtextxy(50, max_y - 20, "An phim bat ky de ket thuc");
getch();
}
```

Chương 4

VECTƠ, MA TRẬN, HỆ PHƯƠNG TRÌNH ĐẠI SỐ TUYẾN TÍNH

Bài số 46 Cho hai vectơ trong tọa độ ba chiều. Lập chương trình tính tổng, hiệu, tích vô hướng và tích của hai vectơ.

Bài giải

Tổng hai vectơ có tọa độ là : x1 + x2 , y1 + y2 , z1 + z2

Hiệu hai vectơ có tọa độ là : x1 - x2 , y1 - y2 , z1 - z2

Tích vô hướng hai vec tơ có toạ độ là : x1 * x2, y1 * y2, z1 * z2

Tích vectơ có tọa độ là :

y1 * z2 - z1 * y2, z1 * x2 - x1 * z2, x1 * y2 - y1 * x2

```
/* CT46 Chuong trinh tinh cac dai luong vecto */
#include <stdio.h>
main()
{
float x1,x2,y1,y2,z1,z2;
printf("\n Chuong trinh tinh vecto toa do 3 chieu\n");
printf("Cho toa do vecto 1 :");
```

```
scanf("%f %f %f",&x1,&y1,&z1);
printf("\n");
printf("Cho toa do vecto 2 :");
scanf("%f %f %f",&x2,&y2,&z2);
printf("\n");
printf(" Vecto 1 + Vecto 2 = %6.2f %6.2f %6.2f\n",
            x1+x2,y1+y2,z1+z2);
printf(" Vecto 1 - Vecto 2 = %6.2f %6.2f %6.2f\n",
            x1-x2,y1-y2,z1-z2);
printf(" Tich vo huong = %6.2f %6.2f %6.2f\n",
         x1*x2,y1*y2,z1*z2);
printf(" Tich vecto = %6.2f %6.2f %6.2f\n",
         y1*z2-z1*y2,z1*x2-x1*z2,x1*y2-y1*x2);
getch();
return;
}
```

Kết quả chương trình CT46 là :

Cho toa do vecto 1 : 12 34 -6.57

Cho toa do vecto 2 : 42.7 24 -6.25

Vecto 1 + vecto 2 = 54.70 58.00 -13.00

Vecto 1 + vecto 2 = -30.70 10.00 -0.50

Tich vo huong = 512.40 816.00 42.19

Tich vecto = -50.50 -213.23 -1163.80

Bài số 47 Cho hai vectơ n chiều

$x = (x_1, x_2, ..., x_n)$

$y = (y_1, y_2, ..., y_n)$

Tính vectơ $z = (z_1, z_2, \ldots, z_n)$ có các thành phần xác định như sau :

$$z_i = \begin{cases} 0 & \text{nếu } 0 < x_i + y_i < A,\ A > 0. \\ x_i + y_i & \text{nếu } x_i + y_i \geq A \\ (x_i + y_i)^2 & \text{nếu } x_i + y_i \leq 0,\ i = 1,2,\ldots,n \end{cases}$$

và tính tích vô hướng của x và z

Bài giải Chương trình CT47 gồm việc nhập dữ liệu hai vectơ **x** và y, một số dương và thử các điều kiện $x_i = y_i$ để rẽ nhánh chương trình.

```
/* CT47 Chuong trinh tinh tich vecto n chieu */
#include <stdio.h>
#include <math.h>
typedef float vt[100];

main()
{
int i,n;
float p,a;
vt x,y,z;
printf("\nChuong trinh tinh tich vecto n chieu\n");
printf(" Cho chieu cua 2 vecto n = ") ;scanf("%d",&n);
printf(" Cho toa do cua 2 vecto\n");
for (i=1;i<=n;i++)
    {
    printf("x[%d] = ",i);
    scanf("%f",&x[i]);
    printf("y[%d] = ",i);
    scanf("%f",&y[i]);
    }
```

```
printf("\n Vao so duong a = ");
scanf("%f",&a);
for (i=1;i<=n;i++)
    {
    if ((x[i]+y[i])<=0) z[i]=pow((x[i]+y[i]),2);
    else if ((x[i]+y[i])<a) z[i]=0;
    else if ((x[i]+y[i])>=a) z[i]=x[i]+y[i];
    printf("z[%d] = %8.2f\n",i,z[i]);
    }
p=0;
for (i=1;i<=n;i++)
p+=x[i]*z[i];
printf("Tich vo huong cua x va z la : %8.2f\n",p);
getch();
return;
}
```

Kết quả ta được :

Chuong trinh tinh tich vecto n chieu

Cho chieu cua 2 vecto n = 3

Cho toa do cua hai vecto

x[1] = 2.75

y[1] = -5,25

x[2] = 28.65

y[2] = 23.75

x[3] = -4

y[3] = 20.15

Vao so duong a = 2

z[1] = 6.25

z[2] = 52.40

z[3] = 16.15

Tich vo huong cua x va z la : 1453.85

Bài số 48 Tính ma trận chuyển vị của ma trận A

Bài giải Cho ma trận A, ma trận chuyển vị có $b_{ij} = a_{ji}$

Dùng hai chu trình for để thay đổi chỉ số hàng và cột và hoán vị chỉ số hàng cột.

```
/* CT48 Chuong trinh tinh ma tran chuyen vi */
#include <stdio.h>
typedef float mt[10][10];

main()
{
int i,jn,m;
float x;
mt a,b;
printf("\n%20c Tinh ma tran chuyen vi cua mot ma tran \n",'');
/* Cach tao blanc : bang so ky tu trong (%20c) duoc in ra */
/* Chu y : so blanc (' ') khong co y nghia va khong duoc qua 3 */
printf(" Cho so chieu cua ma tran\n");
printf(" So hang m = ") ;scanf("%d",&m);
printf(" So cot n = ") ;scanf("%d",&n);
printf(" Vao ma tran a[m,n]\n");
```

```
for (i=1;i<=m;i++)
    for (j=1;j<=n;j++)
        {
        printf(" a[%d][%d] = ",i,j);
        scanf("%f",&x);
        a[i][j] = x;
        }
for (i=1;i<=n;i++)
    for (j=1;j<=m;j++)
        b[i][j]=a[j][i];
for (i=1;i<=m;i++)
    {
    for (j=1;j<=n;j++)
    printf("a[%d][%d] = %8.2f ",i,j,a[i][j]);
    printf("\n");
    }
printf("\n\n");
for (i=1;i<=n;i++)
    {
    for (j=1;j<=m;j++)
    printf("b[%d][%d] = %8.2f ",i,j,b[i][j]);
    printf("\n");
    }
/* Cach in kieu khac */
printf("Ma tran A\n");
for (i=1;i<=m;++i)
    for (j=1;j<=n;++j)
        printf("%8.2f%c",a[i][j],j==n?'\n':' ');
```

```
printf("\n\n");
printf("Ma tran chuyen vi B\n");
for (i=1;i<=n;++i)
    for (j=1;j<=m;++j)
        printf("%8.2f%c",b[i][j],j==m?'\n':'');
getch();
return;
}
```

Kết quả chương trình CT48 như sau :

So hang m = 3

So cot n = 2

Vao ma tran a[m,n]

a[1][1] = 12

a[1][2] = 34

a[2][1] = 45

a[2][2] = 78

a[3][1] = -95

a[3][2] = 67

a[1][1] = 12.00   a[1][2] = 34.00

a[2][1] = 45.00   a[2][2] = 78.00

a[3][1] = 67.00   b[3][2] = 67.00

b[1][1] = 12.00   b[1][2] = 45.00   b[1][3] = -95.00

b[2][1] = 34.00   b[2][2] = 78.00   b[2][3] = 67.00

ma tran A

12.00   34.00

45.00   78.00

-95.00   67.00

Ma tran chuyen vi B

12.00 45.00 -95.00

34.00 78.00 67.00

Bài số 49 Tìm số nhỏ nhất trong các phần tử của một ma trận.

Bài giải Chương trình gồm việc vào dữ liệu ma trận m hàng **n** cột và dùng hai câu lệnh for để thay đổi chỉ số hàng i, cột j. Thuật toán tìm số lớn nhất max và nhỏ nhất min tương tự như thuật toán trong chương trình CT27.

```
/* CT49 Chuong trinh tinh Min va Max trong ma tran */
#include <stdio.h>
float a[10][10];

main()
{
int i,j,m,n;
float max,min,x;
printf("\nChuong trinh tim MAX va MIN trong ma tran\n");
printf("Cho hang cua ma tran m = ");
scanf("%d",&m);
printf("Cho cot cua ma tran n = ");
scanf("%d",&n);
printf("\nVao ma tran A\n");
for (i=1;i<=m;i++)
    for (j=1;j<=n;j++)
        {
        printf("A[%d,%d] = ",i,j);
        scanf("%f",&x);
```

```
        a[i][j]=x;
        }
max = a[1][1];
min = a[1][1];
for (i=1;i<=m;i++)
    for (j=1;j<=n;j++)
        {
        if (max>a[i][j])
            max = max;
        else
            max = a[i][j];
        if (min<a[i][j])
            min = min;
        else
            min = a[i][j];
    }
printf(" So lon nhat MAX = %5.2f\n",max);
printf(" So nho nhat MIN = %5.2f\n",min);
getch();
return;
}
```

Ta được kết quả chương trình CT49 như sau :

Chuong trinh tim MAX va MIN trong ma tran

Cho hang cua ma tran m = 2

Cho cot cua ma tran n = 2

Vao ma tran A

A[1,1] = 15

A[1,2] = 35

A[2,1] = 76

A[2,2] = 28

So lon nhat MAX = 76.00

So nho nhat MIN = 15.00

Bài số 50 Tính tổng của hai ma trận

Bài giải Tổng của hai ma trận C = A + B có các phần tử :

$$c_{ij} = a_{ij} + b_{ij}$$

```
/* CT50 Chuong trinh tinh tong hai matran */
#include <stdio.h>
typedef float mt[10][10];

main()
{
int i,j,k,n,m;
float x;
mt a,b,c;
/* Vao ma tran A */
printf(" Cho cap cua ma tran n = " );scanf("%d",&n);
printf(" m = " );scanf("%d",&m);
for (i=1;i<=n;i++)
    for (j=1;j<=m;j++)
        {
        printf("\n a[%d][%d] = ",i,j);
        scanf("%f",&x);
```

```
    a[i][j] = x ;
    }
printf("\n");
/* Vao ma tran B */
for (i=1;i<=n;i++)
    for (j=1;j<=m;j++)
        {
        printf("\n b[%d][%d] = ",i,j);
        scanf("%f",&x);
        b[i][j] = x;
        }
printf("\n");
/* Tinh tong a+b */
for (i=1;i<=n;i++)
    for (j=1;j<=m;j++)
        {
        c[i][j] = a[i][j]+b[i][j];
        }
/* Hien thi ma tran A tren man hinh */
printf(" Ma tran A\n\n");
for (i=1;i<=n;i++)
    {
    for (j=1;j<=m;j++)
        printf("%4.2f ",a[i][j]);
    printf("\n");
    printf("\n");
    }
```

```
/* Hien thi ma tran B tren man hinh */
printf(" Ma tran B\n\n");
for (i=1;i<=n;i++)
    {
    for (j=1;j<=m;j++)
        printf("%4.2f ",b[i][j]);
    printf("\n");
    printf("\n");
    }
printf("\n");
/* Hien thi ma tran tong tren man hinh */
printf(" Ma tran ket qua\n\n");
for (i=1;i<=n;i++)
    {
    for (j=1;j<=m;j++)
        printf("%4.2f ",c[i][j]);
    printf("\n");
    printf("\n");
    }
getch();
return;
}
```

Kết quả chương trình CT50 như sau :

Cho cap cua ma tran n = 2 m = 2

a[1][1] = 23

a[1][2] = 55

a[2][1] = 10

a[2][2] = 25.75

b[1][1] = - 34.25

b[1][2] = 37.68

b[2][1] = 75.25

b[2][2] = 56.25

Ma tran A

23.00 55.00

10.00 25.75

Ma tran B

-34.25 37.68

75.25 56.25

Ma tran ket qua

-11.25 92.68

85.25 82.00

Bài số 51 Tính ma trận tích C = A.B

Bài giải Các phần tử c_{ij} của ma trận tích được xác định như sau :

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}.$$

Việc tính c_{ij} đưa về bài toán tính tổng của một dãy số. Cần phải thực hiện ba chu trình for để biến đổi các chỉ số i,jk

```
/* CT51 Chuong trinh nhan ma tran tong quat */
#include <stdio.h>
typedef float mt[10][10];
main()
{
int i,jk,m,n,p;
float x;
```

```
mt a,b,c;
/* Vao ma tran A m cot va n hang */
printf(" Vao ma tran A m cot n hang :\n\n");
printf(" Cho so cot cua ma tran m = " );
scanf("%d",&m);
printf(" Cho so hang cua ma tran n = " );
scanf("%d",&n);
for (i=1;i<=m;i++)
    for (j=1;j<=n;j++)
        {
        printf("\n a[%d][%d] = ",i,j);
        scanf("%f",&x);
        a[i][j] = x;
        }
printf("\n");
/* Vao ma tran B n cot p hang */
printf(" Vao ma tran B n cot p hang :\n\n");
printf(" Cho so cot cua ma tran n = " );
scanf("%d",&n);
printf(" Cho so hang cua ma tran p = " );
scanf("%d",&p);
for (i=1;i<=n;i++)
    for (j=1;j<=p;j++)
        {
        printf("\n b[%d][%d] = ",i,j);
        scanf("%f",&x);
        b[i][j] = x;
        }
```

```
printf("\n");
/* Tinh tich a*b */
for (i=1;i<=m;i++)
    for (j=1;j<=p;j++)
        {
        c[i][j] =0;
        for (k=1;k<=n;k++)
            c[i][j] += a[i][k]*b[k][j];
        }
/* Hien thi ma tran A tren man hinh */
printf(" Ma tran A\n\n");
for (i=1;i<=m;i++)
    {
    for (j=1;j<=n;j++)
        printf("%4.2f ",a[i][j]);
    printf("\n");
    printf("\n");
    }
/* Hien thi ma tran B tren man hinh */
printf(" Ma tran B\n\n");
for (i=1;i<=n;i++)
    {
    for (j=1;j<=p;j++)
    printf("%4.2f ",b[i][j]);
    printf("\n");
    printf("\n");
    }
printf("\n");
```

```
/* Hien thi ma tran tich tren man hinh */
printf(" Ma tran ket qua\n\n");
for (i=1;i<=m;i++)
    {
    for (j=1;j<=p;j++)
        printf("%4.2f ",c[i][j]
    printf("\n");
    printf("\n");
    }
getch();
return;
}
```

Kết quả chương trình CT51 như sau :

Vao ma tran A m cot n hang :

Cho so cot cua ma tran m = 2

Cho so hang cua ma tran n = 3

Vao ma tran B n cot p hang :

Cho so cot cua ma tran n = 3

Cho so hang cua ma tran p = 2

Ma tran A

12.00 34.00 56.00

78.00 91.00 45.00

Ma tran B

67.00 45.00

23.25 57.75

45.00 91.23

Ma tran ket qua

4114.50 7612.38

9366.75 12870.60

Bài số 52 Tính ma trận tích C = A.B số phần tử cố định

Bài giải Đây là một trường hợp riêng của chương trình nhân ma trận tổng quát CT51. Áp dụng thuật toán tương tự với số hàng và cột cố định ta được kết quả của chương trình CT52.

```
/* CT52 Chuong trinh nhan ma tran so phan tu la co dinh */
#include <stdio.h>
main()
{
float a[3][2],b[2][4],c[3][4],x;
int i,j,k;
/* Vao ma tran A */
printf(" Vao ma tran A :\n");
for (i=1;i<=3;++i)
    for (j=1;j<=2;++j)
        {
        printf("\n a[%d][%d] = ",i,j);
        scanf("%f",&x);
        a[i][j] = x;
        }
/* Vao ma tran B */
printf(" Vao ma tran B :\n");
for (i=1;i<=2;++i)
    for (j=1;j<=4;++j)
```

```
        {
        printf("\n b[%d][%d] = ",i,j);
        scanf("%f",&x);
        b[i][j] = x;
        }
/* Tinh tich a*b */
for (i=1;i<=3;i++)
    for (j=1;j<=4;j++)
        {
        c[i][j] =0;
        for (k=1;k<=2;k++)
            c[i][j] += a[i][k]*b[k][j];
        }
/* Hien thi ma tran a tren man hinh */
printf(" Ma tran a\n");
for (i=1;i<=3;i++)
    {
    for (j=1;j<=2;j++)
        printf("%4.2f ",a[i][j]);
    printf("\n");
    printf("\n");
    }
/* Hien thi ma tran b tren man hinh */
printf(" Ma tran b\n");
for (i=1;i<=2;i++)
    {
    for (j=1;j<=4;j++)
        printf("%4.2f ",b[i][j]),
```

```
    printf("\n");
    printf("\n");
    }
printf("\n");
/* Hien thi ma tran tich tren man hinh */
printf(" Ma tran ket qua\n");
for (i=0;i<2;i++)
    {
    for (j=1;j<4;j++)
        printf("%4.2f ",c[i][j]);
    printf("\n");
    printf("\n");
    }
getch();
return;
}
```

Ma tran a

24.00 45.00

67.00 -57.50

34.60 47.45

Ma tran b

45.20 34.56 20.45 65.12

25.00 68.12 23.00 8.75

Ma tran ket qua

25.00 68.12 23.00

2209.80 3894.84 1525.80

Bài số 53 Tính tích của hai ma trận với các phần tử là phức.

Bài giải Tổ chức một cấu trúc số phức có trường là phần thực reel và phần số ảo imag. Xây dựng hàm MT_tich để tính tích hai ma trận với các phần tử phức.

```
/* CT53 Chuong trinh tinh toan hai ma tran phuc dung pointer */
#define N 3
#define P 2
#define Q 4
typedef struct
    {double reel;
    double imag;
    }sophuc;

main()
{
void MT_tich(sophuc*,sophuc*,sophuc*,int,int,int);
sophuc a[N][P],b[P][Q],c[N][Q];
int i,j;
/* Khoi tao ma tran A */
for (i=0;i<N;i++)
    for (j=0;j<P;j++)
        {
        a[i][j].reel = i+j;
        a[i][j].imag = i+2*j;
        }
/* Khoi tao ma tran b */
for (i=0;i<P;i++)
    for (j=0;j<Q;j++)
```

```
        { b[i][j].reel  = i+j;
        b[i][j].imag = i+2*j;
        }

/* Tinh tich a*b */
MT_tich((sophuc*) &a,(sophuc*) &b,(sophuc*) &c,N,P,Q);

/* Hien thi ma tran A tren man hinh */
printf(" Ma tran A\n");
for (i=0;i<N;i++)
    {
    for (j=0;j<P;j++)
        printf("(%4.1f)+j(%4.1f) ",a[i][j].reel,a[i][j].imag);
    printf("\n");
    }

/* Hien thi ma tran B tren man hinh */
printf(" Ma tran B\n");
    for (i=0;i<N;i++)
        {
        for (j=0;j<Q;j++)
            printf("(%4.1f)+j(%4.1f) ",b[i][j].reel,b[i][j].imag);
        printf("\n");
        }
printf("\n");

/* Hien thi ma tran tich tren man hinh */
printf(" Ma tran Tich\n");
for (i=0;i<N;i++)
```

```
    {
    for (j=0;j<Q;j++)
        printf("(%4.1f)+j(%4.1f) ",c[i][j].reel,c[i][j].imag);
    printf("\n");
    }
printf("\n");
scanf() ; /* Dung man hinh de xem ket qua */
return 0 ;
}

void MT_tich(sophuc* a,sophuc* b,sophuc* c,int n,int p,int q)
{
void Tich();
int i,j,k;
sophuc s,pr;
sophuc *aik,*bkj*cij
cij = c;
for (i=0;i<n;i++)
    for (j=0;j<q;j++)
        {
        aik = a + i*p;
        bkj = b + j
        s.reel = 0;
        s.imag = 0;
        for (k=0;k<p;k++)
            {
            Tich(aik,bkj&pr);
            s.reel += pr.reel;
```

```
            s.imag += pr.imag;
            aik++;
            bkj += q;
            }
        *(cij++) = s;
    }
} /* cua MT_tich */

void Tich(x,y,prod)
sophuc *x,*y,*prod;
{
prod->reel = x->reel * y->reel - x->imag * y->imag;
prod->imag = x->reel * y->imag + x->imag * y->reel;
}
```

Ta được kết quả chương trình CT53 như sau :

Ma tran A

(0.0) + j(0.0)   (1.0) + j(2.0)

(1.0) + j(1,0)   (2.0) + j(3.0)

(2.0) + j(2.0)   (3.0) + j(4.0)

Ma tran B

(0.0) + j(0.0)   (1.0) + j(2.0)   (2.0) + j(4.0)   (3.0) + j(6.0)

(1.0) + j(1.0)   (2.0) + j(3.0)   (3.0) + j(5.0)   (4.0) + j(7.0)

(-1.0) + j(3.0)   (-4.0) + j(7.0)   (-7.0) + (11.0)   (-10.0) + j(15.0)

Ma tran Tich

(-1.0) + j(3.0)   (- 4.0) + j(7.0)   (-7.0) + j(11.0)   (-10.0) + j(15.0)

(-1.0) + j(5.0)   (-6.0) + j(15.0)   (-11.0) + j(25.0)   (-16.0) + j(35.0)

(-1.0) + j(7.0)   (-8.0) + j(23.0)   (-15.0) + j(39.0)   (-22.0) + j(55.0)

Bài số 54 Nhân hai ma trận C = A.B kiểu FILE.

Bài giải Các thông tin về ma trận được tổ chức thành từng tệp (file)

Dùng hàm fopen (fa,"w") để mở tệp và ghi các dữ liệu là các phần tử của ma trận a_{ij} và b_{ij} .

Dùng hàm fclose (fp) để đóng tệp. Giá trị của hàm fopen được gán cho biến fp cần được khai báo trước theo kiểu con trỏ FILE * fp

Chương trình CT54 bao gồm việc vào ma trận A,B dạng tệp dữ liệu. Xây dựng hàm vào ma trận mtin, ra ma trận mtout, nhân ma trận nhan.

```
/* CT54 Chuong trinh nhan hai ma tran kieu FILE */
#include <stdio.h>
main()
{
FILE *fp;
float x,a[3][2],b[2][4],c[3][4];
char fa[40]; /* Ten tep dai nhat 40 ky tu */
int i,j;
printf("\nVao ma tran A\n");
mtin(a,3,2);
printf("\nVao ma tran B\n");
mtin(b,2,4);
printf("\nVao ten tep du lieu : ");
scanf("%s",fa);
fp = fopen(fa,"w");
for (i=1;i<=3;i++)
    for (j=1;j<=2;j++)
        fprintf(fp,"%f\n",a[i][j]);
fclose(fp);
```

```
fp = fopen(fa,"a");
for (i=1;i<=3;i++)
    for (j=1;j<=4;j++)
        fprintf(fp,"%f\n",b[i][j]);
fclose(fp);
fp = fopen(fa,"r");
for (i=1;i<=3;i++)
    for (j=1;j<=2;j++)
        {
        fscanf(fp,"%f",&x);
        a[i][j] = x;
        }
for (i=1;i<=2;i++)
    for (j=1;j<=4;j++)
        {
        fscanf(fp,"%f",&x);
        b[i][j] = x;
        }
nhan(a,b,c,3,2,4);
fprintf(stdout,"\n Ma tran A\n\n");
mtout(a,3,2);
printf("\n Ma tran B\n\n");
mtout(b,2,4);
fprintf(stdout,"\n Ma tran Tich\n\n");
mtout(c,3,4);
getch();
return;
}
```

```
/* Cac ham */

mtin(pa,m,n)
float *pa;
int m,n;
{
float x;
int k=m*n,t,i,j;
for(t=0;t<k;t++)
    {
    i = t/n;
    j = t - i*n;
    printf("\nPhan tu %d,%d = ",i+1,j+1);
    scanf("%f",&x);
    *(pa+t) = x;
    }
return;
}

mtout(pa,m,n)
float *pa;
int m,n;
{
int k=m*n,t,s;
for(t=0;t<k;++t)
    {
    s = (t+1)%n;
    printf("%4.1f%c",*(pa+t),s?' ':'\n');
    }
```

```
printf("\n\n");
return;
}

nhan(pa,pb,pc,m,n,k)
float *pa,*pb,*pc;
int m,n,k;
{
int i,jt;
for(i=0;i<m;++i)
    for(j=0;j<k;++j)
        {
        *(pc+i*k+j) =0;
        for(t=0;t<n;++t)
            *(pc+i*k+j) +=(*(pa+i*n+t))*(*(pb+t*k+j));
        }
return;
}
```

Kết quả chương trình CT54 như sau :

Vao ma tran A

Phan tu 1,1 = 1

Phan tu 1,2 = 3

Phan tu 2,1 = 5

Phan tu 2,2 = 6

Phan tu 3,1 = 0

Phan tu 3,2 = 2.5

Vao ma tran B

Phan tu 1,1 = 2

Phan tu 1,2 = 0

Phan tu 1,3 = 4

Phan tu 1,4 = 2.4

Phan tu 2,1 = 3

Phan tu 2,2 = 6

Phan tu 2,3 = -4

Phan tu 2,4 = 5

Ma tran A

1.0 3.0

5.0 6.0

0.0 2.5

Ma tran B

2.0 0.0 4.0 2.4

3.0 6.0 -4.0 5.0

Ma tran tich

11.0 18.0 -8.0 17.4

28.0 36.0 -4.0 42.0

7.5 15.0 -10.0 12.5

Bài số 55 Chương trình vào ma trận tổng quát dùng FILE

Bài giải Chương trình CT55 xử lý vào hai ma trận A và B con trỏ kiểu FILE dùng fopen để mở file ghi dữ liệu và fclose để đóng file.

```
/* CT55 Chuong trinh vao ma tran tong quat voi file */
#include <stdio.h>
typedef float mt[50][50];
```

```
void main()
{
FILE *fsl1,*fs12,*fs13,*fopen(),*tg;
mt a,b,c;
float x;
int i,jm,n;
/* Mo file ghi du lieu */
printf("\nVao so hang m, so cot n cua ma tran A\n");
printf("m = ");scanf("%d",&m);
printf("n = ");scanf("%d",&n);
for (i=1;i<=m;i++)
    for (j=1;j<=n;j++)
        {
        printf("\n a[%d][%d] = ",i,j);
        scanf("%f",&x);
        a[i][j] = x;
        }
fsl1 = fopen("mata.dat","w");
for (i=1;i<=m;i++)
    for (j=1;j<=n;j++)
    fprintf(fsl1,"%f\n",a[i][j]);/* Dac ta \n doc du lieu trong file */
fclose(fsl1);
getch();
/* Doc du lieu tu file */
tg = fopen("mata.dat","r");
for (i=1;i<=m;i++)
    for (j=1;j<=n;j++)
```

```
        {
        fscanf(tg,"%f",&x);
        a[i][j] = x;
        }
fsl2,= fopen("matb.dat","w");
printf("\nVao so hang n, so cot p cua ma tran B\n");
printf("n = ");scanf("%d",&n);
printf("p = ");scanf("%d",&p);
for (i=1;i<=n;i++)
    for (j=1;j<=p;j++)
        {
        printf("\n b[%d][%d] = ",i,j);
        scanf("%f",&x);
        b[i][j] = x;
        }
for (i=1;i<=n;i++)
    for (j=1;j<=p;j++)
    fprintf(fsl1,"%f\n",b[i][j]);/* Dac ta \n doc du lieu trong file */
fclose(fsl2);
getch();
return;
}
```

Kết quả ta được :

Vao so hang m, so cot n cua ma tran A

m = 2

n = 2

a[1][1] = 1

a[1][2] = 2

a[2][1] = 3

a[2][2] = 4

vao so hang n, so cot p cua ma tran B

n = 2

p = 2

b[1][1] = 0

b[1][2] = -1

b[2][1] = 4

b[2][2] = 2

Bài số 56 Tính định thức của một ma trận.

Bài giải Để tìm giá trị của định thức D ta biến đổi ma trận sao cho các phần tử dưới đường chéo chính bằng không, do đó giá trị của định thức bằng tích của các phần tử trên đường chéo này

$$D(A) = (-1)^p a_{11}\, a_{22}\, a_{33} \ldots a_{nn}$$

với p là tổng các phép hoán vị đã được thực hiện.

```
/* CT56 Chuong trinh tinh dinh thuc cap n */
#include <stdio.h>
#define TRUE 1
#define FALSE 0

main()
{
int i,jk,n,ok1,ok2;
float d,c,x,a[10][10];
```

```
printf(" Tinh dinh thuc cap n\n");
printf(" Vao bac cua dinh thuc n = ");
scanf("%d",&n);
printf(" Vao ma tran A\n");
for (i=1;i<=n;i++)
    for (j=1;j<=n;j++)
        {
        printf("\n a[%d][%d] = ",i,j);
        scanf("%f",&x);
        a[i][j] = x;
        }
/* Hien thi ma tran A tren man hinh */
printf(" Ma tran A\n");
for (i=1;i<=n;i++)
    {
    for (j=1;j<=n;j++)
        printf("%4.2f ",a[i][j]);
    printf("\n");
    }
d=1;
i=1;
ok2=TRUE;
while (ok2 && (i<=n))
    {
    if (a[i][i]==0)
        {
        ok1=TRUE;
        k=i+1;
```

```
while (ok1 && (k<=n))
    {
    if (a[k][i]!=0)
        {
        for (j=i;j<=n;j++)
            {
            /* Doi dong i cho dong k */
            c=a[i][j];
            a[i][j]=a[k][j];
            a[k][j]=c;
            }
        d = -d;
        ok1=FALSE;
        }
    else k++;
    if (k>n)
        {
        printf(" MA TRAN SUY BIEN\n");
        ok2=FALSE;
        d=0;
        }
    }
if (a[i][i]!= 0)
   {
   c=a[i][i];
   for (j=i+1;j<=n;j++) a[i][j]=a[i][j]/c;
   for (k=i+1;k<=n;k++)
       {
```

```
        c=a[k][i];
        for (j=i+1;j<=n;j++) a[k][j]=a[k][j]-a[i][j]*c;
        }
    }
    i=i+1;
    }
    while(ok2 && (i<=n))
        {
        if (a[i][i]!= 0)
            {
            c=a[i][i];
            for (j=i+1;j<=n;j++) a[i][j]=a[i][j]/c;
            for (k=i+1;k<=n;k++)
                {
                c=a[k][i] ;
                for (j=i+1;j<=n;j++) a[k][j]=a[k][j]-a[i][j]*c;
                }
            i++;
            }
            }
            if (ok2)
            {
                for (i=1;i<=n;i++)
                    d*= a[i][i];
            printf(" GIA TRI DINH THUC D = %10.2f\n",d);
            }
     }
getch();
```

```
return;
}
```

Kết quả của chương trình CT56 như sau :

Tinh dinh thuc cap n

Vao bac cua dinh thuc n = 2

Vao ma tran A

a[1][1] = 2

[1][2] = 4.5

a[2][1] = 3.6

a[2][2] = -1.5

Ma tran A

2.00 4.50

3.60 -1.50

GIA TRI DINH THUC D = -19.20

Bài số 57 Tính ma trận nghịch đảo

Bài giải Nghịch đảo của ma trận A ký hiệu là A^{-1} sao cho $A.A^{-1} = 1$.

Có nhiều thuật toán khác nhau để tính ma trận nghịch đảo. Một trong các thuật toán là :

Nếu bên cạnh ma trận A có ma trận đơn vị, chia các hàng của ma tận này cho cùng một hệ số rồi trừ đi một số hàng khác để ma trận A trở thành ma trận đơn vị thì kết quả ma trận mới tạo nên là nghịch đảo của ma trận A.

```
/* CT57 Chuong trinh tinh ma tran nghich dao */
#include <stdio h>
typedef float mt[10][10];
```

```
void main()
{
int t,t1,i,j,k,n;
float c,x;
mt a,b;
printf(" TINH MA TRAN NGHICH DAO\n");
printf(" Vao bac cua ma tran n = ");
scanf("%d",&n);
printf("Vao ma tran ban dau a\n");
for (i=1;i<=n;i++)
    for (j=1;j<=n;j++)
        {
        printf("a[%d][%d] = ",i,j);
        scanf("%f",&x);
        a[i][j] = x;
        }
printf("\n");
for (i=1;i<=n;i++)
    for (j=n+1;j<=2*n;j++)
        {
        if (j==i+n) a[i][j] = 1;
        else a[i][j] = 0;
        }
i=1;
t1=1;
while (t1 && (i<=n))
    {
    if (a[i][i]==0)
```

```
{
t=1;
k=i+1;
while (t && (k<=n))
    {
    if (a[k][i]!=0)
        {
        for (j=1;j<=2*n;j++)
            {
            c = a[i][j];
            a[i][j] = a[k][j];
            a[k][j]     c;
            }
        t=0;
        }
    else k=k+1;
    if (k==n+1)
        {
        if (a[i][k-1]==0)
            {
            printf(" Ma tran suy bien \n");
            t1=0;
            }
        }
    i++;
    }
    if (a[i][i]!=0)
        {
```

```
          c = a[i][i];
          for (j=i; j<=2*n; j++)
              a[i][j] = a[i][j]/c;
          }
      for (k=1;k<=n;k++)
          {
          if (k!=i)
              {
              c = a[k][i];
              for (j=i;j<=2*n;j++)
                  a[k][j]=a[k][j]-a[i][j]*c;
              }
          }
  i++;
  }
while(t1 && (i<=n))
  {
  if (a[i][i]!=0)
      {
      c = a[i][i];
      for (j=i;j<=2*n;j++)
          a[i][j] = a[i][j]/c;
      }
      for (k=1;k<=n;k++)
          {
          if (k!=i)
              {
              c = a[k][i];
```

```
            for (j=i;j<=2*n;j++)
                a[k][j]=a[k][j]-a[i][j]*c;
            }
        }
    i++;
  }
  if (t1)
    {
    printf(" MA TRAN KET QUA\n") ;
    for (i=1 ;i<=n ;i++)
      {
      for (j=n+1 ;j<=2*n ;j++)
        printf("a[%d][%d] = %10.2f ",i,j,a[i][j]) ;
      printf("\n") ;
      }
    }
  }
getch();
return;
}
```

Kết quả chương trình CT57 như sau :

TINH MA TRAN NGHICH DAO

Vao bac cua ma tran n = 2

Vao ma tran ban dau a

a[1][1] = 2

a[1][2] = 1.5

a[2][1] = 3.6

a[2][2] = -4.3

MA TRAN KET QUA

a[1][3] = 0.31 a[1][4] = 0.11

a[2][3] = 0.26 a[2][4] = -0.14

Bài số 58 Giải hệ phương trình đại số tuyến tính bậc n

Bài giải Có nhiều thuật toán khác nhau để giải hệ phương trình đại số bậc n. Phương pháp khử ẩn Gauss như sau :

Dùng phương trình đầu tiên để khử x_1 trong n - 1 phương trình còn lại.

Nếu $a_{11} \neq 0$ ta tính n - 1 nhân thức $m_i = \frac{a_{i1}}{a_{11}}$, i = 2, 3, ...,n

Nhân phương trình đầu với m_i và trừ vào phương trình i với i = 2, 3, ..., n. Tiếp theo ta được các hệ số của phương trình :

$$a_{ij}^{(1)} = a_{ij} - m_i a_{1j} , \qquad i = 2,3,...,n.$$

$$b_i^{(1)} = b_i - m_i b_1 , \qquad j = 1,2,...,n.$$

trong đó $a_{i1}^{(1)} = 0$, ẩn số x_1 đã bị khử trong n - 1 phương trình của hệ thống ban đầu .

Phương trình thứ hai đã biến đổi trong bước đầu tiên được dùng để khử ẩn x_2 trong n - 2 phương trình còn lại.

Ở bước k ẩn số x_k được khử với nhân thức

$$m_i^{(k-1)} = \frac{a_{ij}^{(k-1)}}{a_{kk}^{(k-1)}} , \qquad i = k+1, k+2, ...,n$$

trong đó $a_{kk}^{(k-1)} \neq 0$ và $a_{ij}^{(k)} = a_{ij}^{(k-1)} - m_i^{(k-1)} a_{kj}^{(k-1)}$, (j = k, ..., n)

$$b^{(k)} = b_i^{(k-1)} - m_i^{(k-1)} b_k^{(k-1)} , \ (i = k+1, ..., n)$$

k lấy giá trị liên tiếp từ 1,2 đến n - 1. Trong giai đoạn k = n - 1 ẩn số x_{n-1} bị khử trong phương trình cuối cùng của hệ phương trình. Cuối cùng ta được một hệ phương trình dạng chéo trong đó phương trình cuối cùng chỉ chứa một ẩn x_n.

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1k}x_k + \ldots + a_{1n}x_n &= b_1 \\
a_{21}^{(1)}x_2 + \ldots + a_{2k}^{(1)}x_k + \ldots + a_{2n}^{(1)}x_n &= b_2^{(1)} \\
a_{kk}^{(k-1)}x_k + \ldots + a_{kn}^{(k-1)}x_n &= b_k^{(k-1)} \\
a_{nn}^{(n-1)}x_n &= b_n^{(n-1)}
\end{aligned}
$$

Hệ phương trình có dạng chéo, ta dễ dàng tính :

$$x_n = b^{(n-1)}/a_{nn}^{(n-1)}$$

$$x_{n-1} = (b_{n-1}^{(n-2)} - a_{n-1,n}^{(n-2)}\, x_n)/a_{n-1,n-1}^{(n-2)}$$

. . . .

$$x_j = (b_j^{(j-1)} - a_{jn}^{(j-1)}x_n) - \ldots - a_{jj+1}^{(j-1)}x_{j+1})/a_{jj}^{(j-1)}$$

$$j = n\text{-}2, \ldots ,3,2,1$$

Chương trình CT58 gồm việc cho bậc phương trình, vào các hệ số a_{ij}, b_j và tìm các điều kiện để hệ phương trình tuyến tính bậc n có nghiệm, nếu không sẽ thông báo hệ vô định, hoặc vô nghiệm và in nghiệm của hệ phương trình.

```
/* CT58 Chuong trinh giai he PTTT bac n */
#include <stdio.h>
typedef float mt[20][20];
typedef float vec[21];

void main()
{
int t,t1,i,j,k,n;
float c,z;
mt a;
vec x;
printf("\n He PTTT bac n ");
```

```
printf("\n Vao bac cua phuong trinh n = ");
scanf("%d",&n);
printf("\n Vao ma tran he so \n ");
for (i=1;i<=n;i++)
    for (j=1;j<=n+1;j++)
        {
        printf("a[%d][%d] = ",i,j);
        scanf("%f",&z);
        a[i][j] = z;
        }
i=1 ;
t1=1 ;
while (t1 && (i<n))
    {
    t=1;
    k=i+1;
    while (t && (k<=n))
        {
        if (a[k][i]!=0)
            {
            for (j=1;j<=n+1;j++)
                {
                c = a[i][j];
                a[i][j] = a[k][j];
                a[k][j] = c;
                }
            t=0;
            }
```

```
        else k=k+1;
        if (k==n+1)
            {
            if (a[i][k]==0)
                printf("\n He vo dinh ");
        else printf("\n He vo nghiem ");
        t1=0;
        }
    }
    if (a[i][i]!=0)
        {
        c = a[i][i];
        for (j=i+1;j<=n+1;j++)
        a[i][j] = a[i][j]/c;
        for (k=i+1;k<=n;k++)
            {
            c = a[k][i];
            for (j=i+1;j<=n+1;j++)
            a[k][j]=a[k][j]-a[i][j]*c;
            }
        }
    i = i+1;
    }
if (t1)
    {
    x[n] = a[n][n+1]/a[n][n];
    for (i=n-1;i>=1;i--)
        {
```

```
        x[i] = a[i][n+1];
        for(j=n;j>=i+1;j--)
            x[i] = x[i]-a[i][j]*x[j] ;
        }
    printf("\n Ma tran ket qua \n");
    for (i=1;i<=n;i++)
        {
            for (j=1;j<=n+1;j++)
                printf("a[%d][%d] = %10.3f ",i,j,a[i][j]);
            printf("\n");
        }
    printf("\n Nghiem cua he PTTT bac n ");
    for(i=1;i<=n;i++)
        printf("\n x[%d] = %10.3f",i,x[i]);
    }
getch(); —
return;
}
```

Ta được kết quả của phương trình CT58 như sau :

a[1][1] = -2.5

a[1][2] = 4.1

a[1][3] = 6.2

a[1][4] = 5.25

a[2][1] = 1.75

a[2][2] = 3.8

a[2][3] = 4

a[2][4] = 1.25

a[3][1] = 12

a[3][2] = 3.4

a[3][3] = 1.7

a[3][4] = 2.5

Ma tran ket qua

a[1][1] =1.750   a[1][2] = 2.171   a[1][3] = 2.286 a[1][4] = 0.714

a[2][1] =12.000   a[2][2] =-22.657 a[2][3] = 1.136 a[2][4] = 0.268

a[3][1] = -2.500 a[3][2] = 9.529   a[3][3] = 1.094 a[3][4] = 4.482

Nghiem cua he PTTT bac n

x[1] = 0.870

x[2] = -4.385

x[3] = 4.097

Chương 5

ĐA THỨC
VÀ NỘI SUY ĐA THỨC

Bài số 59 Tính giá trị của một đa thức tại điểm x_0 theo thuật toán Horner.

Bài giải Cho đa thức $P(x) = a_0x^n + a_1x^{n-1} + \ldots + a_{n-1}x + a_n$. Cho bậc của đa thức, cần đưa các dữ liệu vào là các hệ số $a_0, a_1, \ldots, a_n$ và x_0.

Tính giá trị của đa thức theo thuật toán Horner

$$P = (\ldots((a_0x + a_1)x + a_2)x + \ldots + a_{n-1})x + a_n$$

với $P_k = P_{k-1}x + a_k$; $k = 1,2,\ldots,n$; $P_0 = a_0$. Xác định hàm DaThuc để tính giá trị của đa thức theo thuật toán Horner.

```
/* CT59 Chuong trinh tinh gia tri da thuc theo Horner */
#include <stdio.h>
#include <conio.h>
#include <ctype.h>
#define MAXHESO 21
typedef float heso[MAXHESO]; /* So cac he so cua da thuc */
int n;
heso a;
float xo;
```

```
float DaThuc(heso, float ,int );

void main()
{
int i;
printf("\n** TINH GIA TRI DA THUC THEO HORNER **\n");
printf("\nCho mot gia tri Xo : ");
scanf("%f",&xo);
printf("\nCho bac cua da thuc n = ");
scanf("%d",&n);
for (i=0;i<=n;i++)
    {
    printf("a[%d] = ",i);
    scanf("%f",&a[i]);
    }
printf("\nGIA TRI DA THUC = %6.3f\n",DaThuc(a,xo,n));
getch();
}

float DaThuc(heso b,float XXo,int n)
{
int i;
float tg;
tg = b[n];
for (i=n;i>1;--i)
tg = b[i-1]+XXo*tg;
return(tg);
}
```

Ta được kết quả của chương trình CT59 như sau :

*** TINH GIA TRI DA THUC THEO HORNER ***

Cho mot gia tri Xo : 1

Cho bac cua da thuc n = 3

a[0] = 2

a[1] = 1.5

a[2] = 4.25

a[3] = 2.75

GIA TRI DA THUC = 8.500

Bài số 60 Tính tổng của hai đa thức.

Bài giải Cho hai đa thức :

$A(x) = a_1x^n + a_2x^{n-1} + \ldots + a_nx + a_{n+1}$ bậc n và

$B(x) = b_1x^m + b_2x^{m-1} + \ldots + b_mx + b_{m+1}$ bậc m ($m \le n$)

Tổng C(x) = A(x) + B(x) là một đa thức bậc lớn nhất là n.

Các hệ số c_k của đơn thức n-k+1 của C(x) là tổng của các đơn thức n-k+1 của A(x) và B(x).

Tổng quát ta được các hệ số c_k, k = 1, ...,n+1 của đa thức C(x) :

i = n-m

nếu $k <= i$, $c_k = a_k$ | k = 1,..., n+1

nếu không $c_k = a_k + b_k$

```
/* CT60 Chuong trinh cong da thuc */
#include <stdio.h>
void main ()
{
float a[20],b[20],c[20];
int i,k,n,np,m,mp;
```

```
printf("\nCong hai da thuc A(x) bac n va B(x) bac m\n");
printf("Cho bac cua da thuc A(x) n = ");
scanf("%d",&n);
printf("Cho bac cua da thuc B(x) m = ");
scanf("%d",&m);
np = n+1;
mp = m+1;
printf("\n Vao cac he so cua da thuc A(x)\n");
for (i=1;i<=np;i++)
    {
    printf("A(%d) = ",i);
    scanf("%f",&a[i]);
    }
printf("\n Vao cac he so cua da thuc B(x)\n") ;
for (i=1;i<=mp;i++)
    {
    printf("B(%d) = ",i);
    scanf("%f",&b[i]);
    }
i = np - mp;
for (k=1;k<=np;k++)
    if (k<=i) c[k] = a[k];
    else c[k] = a[k] + b[k-1];
printf(" Da thuc A(x) la \n") ;
for (i=1;i<=np;i++)
    printf("A(%d) = %4.2f,",i,a[i]);
printf("\n");
printf(" Da thuc B(x) la \n");
```

```
for (i=1;i<=mp;i++)
    printf("B(%d) = %4.2f,",i,b[i]);
printf("\n");
printf(" Da thuc tong C(x) la \n");
for (k=1;k<=np;k++)
    printf("c(%d) = %4.2f,",k,c[k]);
printf("\n");
getch();
return;
}
```

Kết quả chương trình CT60 như sau :

Cong hai da thuc A(x) bac n va B(x) bac m

Cho bac cua da thuc A(x) n \= 2

Cho bac cua da thuc B(x) m = 2

Vao cac he so cua da thuc A(x)

A(1) = 1

A(2) = 2

A(3) = 3

Vao cac he so da thuc B(x)

B(1) = 4

B(2) = 5

B(3) = 6

Da thuc A(x) la

A(1) = 1.00, A(2) = 2.00, A(3) = 3.00,

Da thuc B(x) la

B(1) = 4.00, B(2) = 5.00, B(3) = 6.00,

Da thuc tong C(x) la

c(1) = 1.00, c(2) = 6.00, c(3) = 8.00,

Bài số 61 Tính tích của hai đa thức.

Bài giải Tích C(x) = A(x).B(x) là một đa thức bậc $l = n + m$.

Hệ số c_k của đơn thức bậc $l - k + 1$ của C(x) là tổng của tất cả tích các hệ số của các đơn thức bậc k_1 ($0 \le k_1 \le n$) của A(x) và k_2 ($0 \le k_2 \le m$) của B(x).

Tổng quát ta xác định bậc l và các hệ số c_k, $k = 1, ..., l + 1$ của C(x)

$$\left[\begin{array}{l} l = n + m \\ \left[\begin{array}{l} \text{nếu } k \le n + 1,\ j = k \\ \text{nếu không } j = n + 1 \\ \text{nếu } k \le m + 1,\ i_o = 1 \\ \text{nếu không } i_o = k - m \\ C_k = \sum_{i=i_o}^{j} a_i b_{k-i+1} \end{array}\right. \quad k = 1, ..., l + 1 \end{array}\right.$$

```
/* CT61 Chuong trinh nhan hai da thuc */
#include <stdio.h>
#define MAXSIZE 50
typedef float vecto[MAXSIZE+1];
void main()
{
int n,np,m,mp;
vecto a,b,c;
int i,io,j,k,lp;
```

```
printf("\nNhan hai da thuc A(x) bac n va B(x) bac m\n");
printf("\nCho bac cua da thuc A(x) n = ");
scanf("%d",&n);
printf("\nCho bac cua da thuc B(x) m = ");
scanf("%d",&m);
np = n+1;
mp = m+1;
printf("\n Vao cac he so cua da thuc A(x)\n");
for (i=1;i<=np;i++)
    {
    printf("A(%d) = ",i);
    scanf("%f",&a[i]);
    }
printf("\n Vao cac he so cua da thuc B(x)\n") ;
for (i=1;i<=mp;i++)
    {
    printf("B(%d) = ",i);
    scanf("%f",&b[i]);
    }
lp = np + mp - 1;
for (k=1;k<=lp;k++)
    {
    if (k<=np) j = k;
    else j = np;
    if (k<=mp) io = 1;
    else io = k - mp + 1;
    c[k] = 0.0;
    for (i=io;i<=j;i++)
```

```
        c[k] = c[k]+a[i]*b[k-i+1];
    }
printf("\nDay he so tich C[k]\n");
for (k=1;k<=lp;k++)
printf("C[%d] = %6.2f\n",k,c[k]);
getch();
return;
}
```

Ta được kết quả chương trình CT61 :

Nhan hai da thuc A(x) bac n va B(x) bac m

Cho bac cua hai da thuc A(x) n = 2

Cho bac cua hai da thuc B(x) m = 2

Vao cac he so cua da thuc A(x)

A(1) = 1

A(2) = 2

A(3) = 3

Vao cac he so cua da thuc B(x)

B(1) = 4

B(2) = 5

B(3) = 6

Day he so tich C[k]

C[1] = 4.00

C[2] = 13.00

C[3] = 28.00

C[4] = 27:00

C[5] = 18.00

Bài số 62 Chia hai đa thức.

Bài giải Cho hai đa thức A(x) và B(x), đa thức tỷ số Q(x) và phần dư R(x) của phép chia A(x) cho B(x).

Tổng quát ta xác định bậc l và các hệ số q_k của đa thức Q(x) như sau :

$$l = n - m$$

$$r_i^{(0)} = a_i \qquad i = 1, \ldots, n$$

$$j_0 = n$$

$$q_k = \frac{r^{k-1}}{b_1}$$

$$\text{nếu } i \le m \text{ , } r_i^{(k)} = r_{i+1}^{(k-1)} - q_k b_{i+1} \quad | \; i = 1, \ldots, j_{k-1} \; | \quad k = 1, \ldots, l+1$$

$$\text{nếu không } r_i^{(k)} = r_{i+1}^{(k-1)}$$

$$j = j_{k-1} - 1$$

```
/* CT62 Chuong trinh chia hai da thuc */
#include <stdio.h>
#define MAXSIZE 50
typedef float vecto[MAXSIZE+1];
void main()
{
int n,np,m,mp;
vecto a,b,q,r;
int i,k;
int l,lp;
int j,jp;
float epsi;
printf("\nChia hai da thuc A(x) bac n va B(x) bac m\n");
printf("\nCho bac cua da thuc A(x) n = ");
```

```
scanf("%d",&n);
printf("\nCho bac cua da thuc B(x) m = ");
scanf("%d",&m);
np = n+1;
mp = m+1;
printf("\n Vao cac he so cua da thuc A(x)\n");
for (i=1;i<=np;i++)
   {
   printf("A(%d) = ",i);
   scanf("%f",&a[i]);
   }
printf("\n Vao cac he so cua da thuc B(x)\n");
for (i=1;i<=mp;i++)
   {
   printf("B(%d) = ",i);
   scanf("%f",&b[i]);
   }
printf("\nCho gia tri epsilon : epsi = ");
scanf("%f",&epsi);
if (mp>1)
   {
   lp = np - mp + 1;
   for (i=1;i<=np;i++)
      r[i] = a[i];
   j = np - 1;
   for (k=1;k<=lp;k++)
      {
      q[k] = r[1]/b[1];
```

```
        for (i=1;i<=j;i++)
          {
          if (i<mp)
            r[i] = r[i+1]-q[k]*b[i+1];
          else r[i] = r[i+1];
          }
        j = j - 1;
        }
        while ((fabs((double)r[1])<(double)(epsi)) && (j>0))
          {
          for (i=1;i<=j;i++)
            r[i] = r[i+1];
          j = j-1;
          }
          if (fabs((double)r[1])<(double)epsi)
            r[1] = 0.0;
          jp = j+1;
    }
  else
    {
    lp = np;
    for (k=1;k<=lp;k++)
      q[k] = a[k]/b[1];
    jp = 1;
    r[1] = 0.0;
    }
  printf("\nIn Ket qua He so Q[k]\n");
```

```
for (k=1;k<=lp;k++)
    printf("Q[%d] = %6.2f\n",k,q[k]);
printf("\nIn Ket qua He so R[i]\n");
for (i=1;i<=p;i++)
    printf("R[%d] = %6.2f\n",i,r[i]);
getch();
return;
}
```

Kết quả chương trình CT62 như sau :

Chia hai da thuc A(x) bac n va B(x) bac m

Cho bac cua da thuc A(x) n = 2

Cho bac cua da thuc B(x) m = 1

Vao cac he so cua da thuc A(x)

A(1) = 1

A(2) = 2

A(3) = 3

Vao cac he so cua da thuc B(x)

B(1) = 4

B(2) = 5

Cho gia tri epsilon : epsi = 0.01

In ket qua He so Q[k]

Q[1] = 0.25

Q[2] = 0.19

In Ket qua He so R[i]

R[1] = 2.06

Bài số 63 Tìm giá trị của hàm f(x) tại điểm x_0 bằng đa thức nội suy Lagrange

Bài giải Cho hàm f(x) xác định bởi cặp các giá trị (x_i, y_i) với i = 1, 2, ..., n + 1. Đa thức nội suy của hàm f(x) có dạng :

$$P(x) = p_1x^n + p_2x^{n-1} + \dots + p_nx + p_{n+1}$$

sao cho $P(x_i) = y_i$, i = 1, 2, ..., n + 1. Cặp (x_i,y_i) gọi là các điểm nội suy. Ta cần phải xác định các hệ số p_1, p_2,...p_{n+1} của đa thức nội suy. Về nguyên tắc giải hệ phương trình tuyến tính có thể xác định các hệ số p_1, ...p_{n+1}. Tuy nhiên trong thực tế ta thường sử dụng nội suy đa thức Lagrange xác định như sau :

$$G_k(x_i) = \begin{cases} 1 \text{ nếu } i=k \\ 0 \text{ nếu } i \neq k \end{cases} \quad i = 1, ..., n+1$$

$$P(x) = y_1G_1(x) + \dots + y_{n+1}G_{n+1}(x)$$

Công thức tính $G_k(x)$ với mọi x là :

$$G_k(x) = \frac{(x-x_1) \dots (x-x_{k-1})(x-x_{k+1}) \dots (x-x_{n+1})}{(x_k-x_1) \dots (x_k-x_{k-1})(x_k-x_{k+1}) \dots (x_k-x_{n+1})}$$

$$= \prod_{\substack{i=1 \\ i \neq k}}^{n+1} \frac{x-x_i}{x_k-x_i} \quad , \quad k = 1, ..., n+1$$

Chương trình CT63 gồm các bước sau :

- Vào dữ liệu các điểm nội suy (x_i,y_i)
- Tính lặp theo công thức Lagrange
- In kết quả

```
/* CT63 Chuong trinh noi suy da thuc Lagrange */
#include <stdio.h>
#include <conio.h>
#include <ctype.h>
```

```
#define NMAX 21
#define TRUE 1
#define FALSE 0
typedef float vecto[NMAX];
int n,i,k,maxKQ;
float po,xo;
vecto x,xx,y,yy,a;
char OK;
void VaoSL(vecto,vecto);
float LapLa(int,vecto,vecto,float);
void InKQ(vecto,vecto);

void main()
{
clrscr();
printf("\n** NOI SUY DA THUC LAGRANGE **");
VaoSL(x,y);
k = 0;
OK = 'C';
do
   {
   printf("\nTinh gia tri cua Y ung voi Xo = ");
   scanf("%f",&xo);
   po = LapLa(n,x,y,xo);
   printf("\nGia tri cua Y = %6.2f",po);
   getch();
   k += 1;
   maxKQ = k;
```

```
    xx[k] = xo;
    yy[k] = po;
    printf("\nCo tinh voi Xo khac nua khong (c/k) : ");
    getchar();
    if (toupper(getchar())=='C')
       OK = 'C';
    else OK = 'K';
    }
 while (OK=='C');
 InKQ(xx,yy) ;
 }

 void VaoSL(vecto x,vecto y)
 {
 int i,t;
 char *OK;
 printf("\nCho ham y = f(x)\n");
 printf("\nSo cap (x,y) nhieu nhat la NMAX = 21");
 printf("\nCho so diem lap n = ");
 scanf("%d",&n);
 for (i=1;i<=n;i++)
    {
    printf("\nX[%d] = ",i);
    scanf("%f",&x[i]);
    printf("\nY[%d] = ",i);
    scanf("%f",&y[i]);
    }
 printf("\Bang so lieu cac cap x va y \n") ;
```

```
printf("\n\tNo\t\tX[i]\t\tY[i]");
for (i=1;i<=n;i++)
    printf("\n\t%d\t\t%4.1f\t\t%4.1f",i,x[i],y[i]);
getch();
/*OK = " ";*/
t = TRUE;
do
    {
    printf("\nCo sua so lieu khong (c/k) : ");
    getchar();
    if (toupper(getchar())=='C')
        {
        printf("\nChi so phan tu can sua i = ");
        scanf("%d",&i);
        printf("\nGia tri moi : X[%d] = ",i);
        scanf("%f",&x[i]);
        printf("\nGia tri moi : Y[%d] = ",i);
        scanf("%f",&y[i]) ;
        }
    if (toupper(getchar()) !='C') t = FALSE;
    }
    while (t);
}

float LapLa(int n,vecto x,vecto y,float xo)
{
int i,k;
float go;
```

```
po = 0.0;
for (k=1;k<=n;k++)
    {
    go = 1;
    for (i=1 ;i<=n ;i++)
        if (i !=k) go *= ((xo-x[i])/(x[k]-x[i]));
    po += (y[k]*go);
    }
return(po);
}

void InKQ(vecto xx,vecto yy)
{
int i,k ;
printf("\nBang so lieu\n");
printf("\n\tNo\t\tX[i]\t\tY[i]");
for (i=1;i<=n;i++)
printf("\n\t%d\t\t%4.1f\t\t%4.1f",i,x[i],y[i]);
getch();
printf("\nKet qua tinh toan\n");
for (k=1;k<=maxKQ;k++)
printf("\n\tX = %6.2f\t\tY = %6.2f",xx[k],yy[k]);
getch();
return;
}
```

Kết quả chương trình CT63 như sau :

Bang so lieu cac cap x va y

No	X[i]	Y[i]
1	1.0	2.0
2	3.0	4.0
3	5.0	6.0
4	7.0	8.0

Co sua so lieu khong (c/k) :

k

Tinh gia tri cua Y ung voi X_o = 0

Gia tri cua Y = 1.00

Co tinh voi Xo khac nua khong (c/k) : k

Bài số 64 Tính giá trị gần đúng của hàm f(x) tại điểm x_0 bằng đa thức nội suy Newton.

Bài giải Đa thức nội suy Newton được định nghĩa như sau :

$$P(x) = \delta[x_1] + \sum_{k=1}^{n} \delta[x_i,...,x_{k+1}]\Phi_k(x)$$

Trong đó $\delta[x_i,...,x_{i+k}] = a_i^{(k)}$ với $1 \le i \le n+1\text{-}k$ và $0 \le k \le n$;

$\delta[x_i,...,x_{n+2-j}] = a_j$ với $1 \le j \le n+1$

là các hệ số của đa thức Newton và được tính như sau :

$a_i^{(o)} = y_i$, i = 1, ..., n+1

$$a_i^{(k)} = \frac{a_{i+1}^{(k-1)} - a_i^{(k-1)}}{x_{i+k} - x_i} \qquad i = 1, \ldots, n+1\text{-}k$$

$$k = 1, \ldots, n$$

$$a_{n+2-k} = a_i^{(k-i)}$$

$$a_j = a_i^{(n)}$$

Viết lại đa thức nội suy Newton theo các hệ số a_i :

$$P(x) = a_1\Phi_n(x) + a_2\Phi_{n-1}(x) + ... + a_n\Phi_1(x) + a_{n+1}$$

với : $\Phi_i(x) = (x-x_1)(x-x_2) ... (x-x_i)$, $i=1,...,n+1$

Chương trình CT64 gồm các bước sau đây :

- Vào dữ liệu là cặp các điểm nội suy (x_i,y_i)
- Dùng hàm tính các hệ số a_i
- Tính lặp theo công thức Newton và in kết quả

```
/* CT64 Chuong trinh noi suy da thuc Newton */
#include <stdio.h>
#include <conio.h>
#include <ctype.h>
#define NPMAX 21
#define TRUE 1
#define FALSE 0
typedef float vecto[NPMAX];
int n,maxKQ;
vecto x,y,a,xx,yy;
float xo;
void VaoSL(vecto,vecto);
void HeSoNEW(int,vecto,vecto,vecto);
float LAPNEW(int,vecto,vecto,float);
void InKQ(vecto,vecto);

void main()
{
int k;
float po;
char OK;
```

```
clrscr();
printf("\n** NOI SUY DA THUC NEWTON **");
VaoSL(x,y);
k = 0;
OK = 'C';
do
   {
   printf("\nTinh gia tri cua Y ung voi Xo = ");
   scanf("%f",&xo);
   po = LAPNEW(n,x,y,xo);
   printf("\nGia tri cua Y = %6.2f",po);
   getch();
   k += 1;
   maxKQ = k;
   xx[k] = xo ;
   yy[k] = po :
   printf("\nCo tinh voi Xo khac nua khong (c/k) : ");
   getchar();
   if (toupper(getchar())=='C') OK = 'C';
   else OK = 'K';
   }
while (OK=='C');
InKQ(xx,yy);
}

void VaoSL(vecto x,vecto y)
{
int i,t;
```

```
printf("\nCho ham y = f(x)\n");
printf("\nSo cap (x,y) nhieu nhat la NMAX = 21");
printf("\nCho so diem lap n = ");
scanf("%d",&n);
for (i=1;i<=n;i++)
    {
    printf("\nX[%d] = ",i);
    scanf("%f",&x[i]);
    printf("\nY[%d] = ",i);
    scanf("%f",&y[i]);
    }
printf("\Bang so lieu cac cap x va y \n");
printf("\n\tNo\t\tX[i]\t\tY[i]");
for (i=1;i<=n;i++)
printf("\n\t%d\t\t%4.1f\t\t%4.1f",i,x[i],y[i]);
getch();
t = TRUE;
do
    {
    printf("\nCo sua so lieu khong (c/k) : ");
    getchar();
    if (toupper(getchar())=='C')
        {
        printf("\nChi so phan tu can sua i = ");
        scanf("%d",&i);                    được
        printf("\nGia tri moi : X[%d] = ",i);
        scanf("%f",&x[i]);
        printf("\nGia tri moi : Y[%d] = ",i);
```

```
        scanf("%f",&y[i]);
        }
    if (toupper(getchar()) !='C') t = FALSE;
    }
while (t);
}

void HeSoNEW(int n,vecto x,vecto y,vecto a)
{
int i,k;
float a1;
a = y;
for (k=1;k<=n-1;k++)
    {
    a1 = a[1];
    for (i=1;i<=n-k;i++)
        a[i] = (a[i+1]-a[i])/(x[i+k]-x[i]);
    a[n-k+1] = a1;
    }
}

float LAPNEW(int n,vecto a,vecto x,float Xo)
{
int k;
float Bo;
Bo = a[1];
for (k=2;k<=n;k++)
    Bo = Bo*(Xo-x[n-k+1])+a[k];
```

```
return(Bo);
}

void InKQ(vecto xx,vecto yy)
{
int i,k;
printf("\nBang so lieu\n");
printf("\n\tNo\t\tX[i]\t\tY[i]");
for (i=1;i<=n;i++)
    printf("\n\t%d\t\t%4.1f\t\t%4.1f",i,x[i],y[i]);
getch();
printf("\nKet qua tinh toan\n");
for (k=1;k<=maxKQ;k++)
    printf("\n\tX = %6.2f\t\tY = %6.2f",xx[k],yy[k]);
getch();
}
```

Ta được kết quả chương trình CT64 như sau :

*** NOI SUY DA THUC NEWTON ***

Cho ham y = f(x)

So cap (x,y)nhieu nhat la NMAX = 21

Cho so diem lap n = 3

X[1] = 1

Y[1] = 2

X[2] = 3

Y[2] = 4

X[3] = 6

Y[3] = 7

Bang so lieu cac cap x va y

No	*X[i]*	*Y[i]*
1	*1.0*	*2.0*
2	*3.0*	*4.0*
3	*6.0*	*7.0*

Tinh gia tri cua Y ung voi X0 = 1.5

Gia tri cua Y = 5.75

Bài số 65 Tìm giá trị đa thức nội suy bằng ma trận Lagrange.

Bài giải Đa thức Lagrange có thể viết dưới dạng ma trận :

$$G_k(x) = (x^n, x^{n-1}, \ldots, x, 1) \begin{pmatrix} C_{1k} \\ C_{2k} \\ \ldots \\ C_{n+1k} \end{pmatrix}$$

Ma trận vuông C_{ij} bậc n + 1 được xác định như sau :

$$C = \begin{pmatrix} C_{11} & C_{12} & \ldots & C_{1n+1} \\ C_{21} & C_{22} & \ldots & C_{2n+1} \\ \ldots & \ldots & \ldots & \ldots \\ C_{n+11} & C_{n+12} & \ldots & C_{n+1\,n+1} \end{pmatrix}$$

được gọi là ma trận Lagrange được xây dựng bởi các giá trị x_1 đến x_{n-1} theo công thức truy hồi. Đa thức nội suy được viết dưới dạng ma trận :

$$P(x) = (x^n, x^{n-1}, \ldots, x, 1)\ .C. \begin{pmatrix} y_1 \\ y_2 \\ \ldots \\ y_{n+1} \end{pmatrix}$$

$$P(x) = (x^n, x^{n-1}, \ldots, x, 1) \begin{pmatrix} p_1 \\ p_2 \\ \ldots \\ p_{n+1} \end{pmatrix}$$

Ma trận hệ số p_i được tính như sau :

$$p_i = \sum_{j=1}^{n+1} C_{ij}\, y_j \qquad i = 1, \ldots, n+1$$

Chương trình CT65 gồm các bước :

- Vào dẫy các điểm nội suy (x_i, y_i)
- Xác định mà trận Lagrange C
- Xác định p
- Tính P(x) và ghi kết quả.

```
/* CT65 Chuong trinh noi suy ma tran Lagrange */

#include <stdio.h>
#include <conio.h>
#include <ctype.h>
#define NMAX 11
#define NPMAX 12
#define TRUE 1
#define FALSE 0
typedef float vecto[NMAX];
typedef float mt[NMAX][NMAX];
int n;
vecto x,y,p;
void VaoSL(vecto,vecto);
```

```
void InMT(int,int,mt);
void MaTranLaX(int,vecto,mt );
void HeSoP(int ,vecto ,vecto ,vecto );
void InKQ(vecto ,vecto ,vecto );

void main()
{
clrscr();
printf("\n** DA THUC NOI SUY CUA HAM MA TRAN
LAGRANGE **");
VaoSL(x,y);
HeSoP(n,x,y,p);
getch();
printf("\nDa thuc noi suy cua HAM F\n");
InKQ(x,y,p);
getch();
}

void VaoSL(vecto x,vecto y)
{
int i,t;
printf("\nCho ham y = f(x)\n");
printf("\nSo cap (x,y) nhieu nhat la NMAX = 11");
printf("\nCho so diem lap n = ");
scanf("%d",&n);
for (i=1;i<=n;i++)
    {
    printf("\nX[%d] = ",i);
    scanf("%f",&x[i]);
```

```
    printf("\nY[%d] = ",i);
    scanf("%f",&y[i]);
    }
printf("\Bang so lieu cac cap x va y \n");
printf("\n\tNo\t\tX[i]\t\tY[i]");
for (i=1;i<=n;i++)
    printf("\n\t%d\t\t%4.1f\t\t%4.1f",i,x[i],y[i]);
getch();
t = TRUE;
do
    {
    printf("\nCo sua so lieu khong (c/k) : ");
    getchar();
    if (toupper(getchar())=='C')
        {
        printf("\nChi so phan tu can sua i = ");
        scanf("%d",&i);
        printf("\nGia tri moi : X[%d] = ",i);
        scanf("%f",&x[i]);
        printf("\nGia tri moi : Y[%d] = ",i);
        scanf("%f",&y[i]);
        }
    if (toupper(getchar()) !='C') t = FALSE;
    }
while (t) ;
```

```
void InMT(int m,int n,mt heso)
{
int i,j;
printf("\nMA TRAN HE SO LAGRANGE\n");
for (i=1;i<m;i++)
   {
   for (j=1;j<n;j++)
      printf("HS[%d][%d] = %4.1f ",i,j,heso[i][j]);
   printf("\n"); /*Xuong dong sau khi het mot hang */
   }
}

void MaTranLaX(int n,vecto x,mt heso)
{
int i,j,k;
float lo,qo,uo,vo;
float phi[NPMAX];
phi[1] = 1.0;
phi[2] = -x[1];
for (i=2;i<=n;i++)
   {
   uo = phi[1];
   for (j=2;j<=i;j++)
      {
      vo = phi[j];
      phi[j] = vo-uo*x[i];
      uo = vo;
      }
```

```
    phi[i+1] = -uo*x[i];
    }
for (k=1;k<=n;k++)
    {
    lo = n;
    for (j=2;j<=n;j++)
        lo = lo*x[k]+(n-j+1)*phi[j];
    qo = phi[1] ;
    heso[1][k] = qo/lo;
    for (j=2;j<=n;j++)
        {
        qo = qo*x[k]+phi[j] ;
        heso[j][k] = qo/lo ;
        }
    }
InMT(jk,heso);
}

void HeSoP(int n,vecto x,vecto y,vecto p)
{
int i,j;
float s;
mt heso;
MaTranLaX(n,x,heso);
for (i=1;i<=n;i++)
    {
    s = 0.0;
    for (j=1;j<=n;j++)
```

```
        s = s+heso[i][j]*y[j];
    p[i] = s;
    }
}

void InKQ(vecto x,vecto y,vecto p)
{
int i,k;
printf("\nBang so lieu\n");
printf("\n\tNo\t\tX[i]\t\tY[i]");
for (i=1;i<=n;i++)
    printf("\n\t%d\t\t%4.1f\t\t%4.1f",i,x[i],y[i]);
getch();
printf("\nKet qua tinh toan\n");
for (k=1;k<=n;k++)
    printf("\n\He so cua don thuc bac %d : %6.2f",n-k,p[k]);
getch() ;
}
```

Ta được kết quả chương trình CT65 :

*** DA THUC NOI SUY CUA HAM MA TRAN LAGRANGE ***

Cho ham y = f(x)

So cap (x,y) nhieu nhat la NMAX = 11

Cho so diem lap n = 3

X[1] = 1

Y[1] = 2

X[2] = 3

Y[2] = 4

X[3] = 5

Y[3] = 6

Bang so lieu cac cap x va y

No	*X[i]*	*Y[i]*
1	*1.0*	*2.0*
2	*3.0*	*4.0*
3	*5.0*	*6.0*

MA TRAN HE SO LAGRANGE

HS[1][1] = 0.1 *HS[1][2] = -0.2* *HS[1][3] = 0.1*

HS[2][1] = -0.1 *HS[2][2] = 1.5* *HS[2][3] = -0.5*

HS[3][1] = 1.9 *HS[3][2] = -1.2* *HS[3][3] = 0.4*

Da thuc noi suy cua HAM F

Ket qua tinh toan

He so cua don thuc bac 2 : 0.00

He so cua don thuc bac 1 : 1.00

He so cua don thuc bac 0 : 1.00

Bài số 66 Tính đạo hàm cấp l của đa thức P(x).

Bài giải Cho đa thức bậc n

$$P(x) = p_1x^n + p_2x^{n-1} + \dots + p_{n+1}$$

Đạo hàm cấp l ($1 \le l \le n$) của P(x) được xác định như sau :

$$P'(x) = n\,p_1x^{n-1} + (n-1)p_2x^{n-2} + \dots + p_n$$

$$P''(x) = n(n-1)\,p_1x^{n-2} + (n-1)(n-2)p_2x^{n-3} + \dots + 2p_{n-1}$$

Tổng quát : $p^{(l)}(x) = d_1x^m + d_2x^{m-1} + \dots + d_{m+1}$

Trong đó $m = n - 1$, $d_k = A^l_{n+1-k}\, p_k$, $k=1,...,m+1$

Ta được bậc m và các hệ số d_k , $k=1,...,m+1$ của đa thức $p^{(l)}(x)$:

$$\begin{cases} m = n - 1 \\ \begin{cases} a_k = \prod_{j=m+2-k}^{n+1-k} j \\ d_k = a_k p_k \end{cases} , \quad k = 1,...,m+1 \end{cases}$$

giá trị của đa thức $p^{(l)}(x_0)$ được tính toán theo sơ đồ Horner.

Chương trình CT66 gồm việc vào dữ liệu các hệ số của đa thức, cấp của đạo hàm và lập một chu trình tính các hệ số của đạo hàmđa thức theo thuật toán tính tích một dãy số.

```
/* CT66 Chuong trinh dao ham da thuc */

#include <stdio.h>
#define MAXSIZE 21
typedef float vec[MAXSIZE];

void main()
{
int i,j,k,l,n,np,mp;
float ao;
vec a,d;
printf("\nChuong trinh tinh dao ham da thuc P(x)\n");
printf("Bac cua da thuc P(x) n = ");
scanf("%d",&n);
printf("Cap cua dao ham l = ");
scanf("%d",&l);
printf("Cho cac he so cua da thuc\n");
```

```
for (i=0;i<=n;i++)
    {
    printf("A[%d] = ",i);
    scanf("%f",&a[i]);
    }
np = n+1;
mp = np-l;
for (k=1;k<=mp;k++)
    {
    ao = 1;
    for (j=mp+1-k;j<=np-k;j++)
    ao = ao*j;
    d[k] = ao*a[k];
    printf("Dao ham cap %d D{%d] = %6.2f\n",l,k,d[k]);
    }
getch();
return;
}
```

Ta được kết quả chương trình CT66 như sau :

Chuong trinh tinh dao ham da thuc P(x)

Bac cua da thuc P(x) n = 2

Cap cua dao ham l = 1

Cho cac he so cua da thuc

A[0] = 2

A[1] = 3

A[2] = 5

Dao ham cap 1 D[1] = 6.00

Dao ham cap 1 D[2] = 5.00

Bài số 67 Tính đạo hàm của đa thức Newton.

Bài giải Cho đa thức Newton

$$P(x) = a_1\Phi_n(x) + ... + a_n\Phi_1(x) + a_{n+1}$$

đạo hàm cấp l của đa thức có thể được tính toán theo thuật toán CT66.

```
/* CT67 Chuong trinh dao ham da thuc Newton */

#include <stdio.h>
#define MAXSIZE 21
typedef float vec[MAXSIZE];

void main()
{
int i,k,l,np;
float uo,xo,a1;
vec a,b,z,x,y;
printf("\nChuong trinh dao ham cap l da thuc Newton\n");
printf(" Cho bac da thuc np = ");
scanf("%d",&np);
printf(" Cho cap cua dao ham l = ");
scanf("%d",&l);
printf(" Cho Xo = ");
scanf("%f",&xo);
```

```
for (i=1;i<=np+1;i++)
    {
    printf("X[%d] = ",i);
    scanf("%f",&x[i]);
    printf("Y[%d] = ",i);
    scanf("%f",&y[i]);
    }
for (i=1;i<=np+1;i++)
    a[i] = y[i];
for (k=1;k<=np-1;k++)
    {
    a1 = a[1];
    for (i=1;i<=np-k;i++)
        {
        a[i] = (a[i+1]-a[i])/(x[i+k]-x[i]);
        a[np-k+1] = a1;
        printf("A[%d] = %6.2f\n",i,a[i]);
        }
    }
for (i=1;i<=np+1;i++)
    b[i] = a[i];
for (i=0;i<=l;i++)
    for (k=2;k<=np-i;k++)
        b[k] = b[k-1]*(xo-x[np+1-k-i]+b[k]);
uo = 1;
for (i=1;i<=l;i++)
    {
    uo = uo*i;
```

```
    z[i] = uo*b[np-i];
    printf("Z[%d] = %6.2f\n",i,z[i]);
    }
getch();
return;
}
```

Kết quả ta được :

Chuong trinh dao ham cap l da thuc Newton

Cho bac da thuc np = 2

Cho cap cua dao ham l = 1

Cho X0 = 2

X[1] = 3

Y[1] = 4

X[2] = 5

Y[2] = 1

X[3] = 6

Y[3] = 2

X[1] = -1.50

Z[1] = -1.50

Chương 6

TÍCH PHÂN SỐ

Chương này trình bày một số thuật toán và chương trình tính tích phân xác định của các hàm giải tích bằng phương pháp số.

Bài số 68 Tính $\int_a^b f(x)dx$ bằng phương pháp Simpson.

Bài giải Cho hàm y = f(x) liên tục trong khoảng [a,b], tích phân xác định của hàm y = f(x) trong khoảng [a,b] được tính gần đúng bằng phương pháp Simpson như sau :

$$\int_a^b f(x)dx = \frac{h}{3}[f(a)+f(b) + 4\sum_{i=1,3}^{m-1} f(a+ih) + 2\sum_{i=2,4}^{m-2} f(a+ih)] + R_m(f)$$

trong đó phần dư $R_m(f) = -\frac{h^4}{180}(b-a)f^4(\xi)$ $,\xi \in [\ a,b\ ]$

số đoạn m tùy ý, bước tích phân h = (b - a)/m

Chương trình CT68 gồm các bước sau :

- Xác định hàm cần lấy tích phân bằng float ham (float x) trong đó hàm $y = e^{-x}\sin x$.

- Vào giá trị cận trên a, cận dưới b và số điểm tính m và xác định bước tích phân h = (b - a)/m.

- Dùng một chu trình for để tính tổng theo công thức Simpson.

```
/* CT68 Chuong trinh tinh tich phan xac dinh pp. Simpson */

#include<stdio.h>
#include<math.h>
float ham(float);

void main()
{
int i,m;
float a,b,x,h,x2,f2,x4,f4,tp;
printf("\n Chuong trinh tinh tich phan pp Simpson");
printf("\n Cho can duoi a = ") ;scanf("%f",&a);
printf("\n Cho can tren b = ") ;scanf("%f",&b);
printf("\n Cho so diem tinh m = ") ;scanf("%d",&m);
h = (b-a)/m;
x2 = a+h;
x4 = a+h/2;
f2 = ham(x2);
f4 = ham(x4) ;
for (i=1;i<m-2;i++)
    {
    x2 = x2+i*h;
    x4 = x4+i*h;
    f2 += ham(x2);
    f4 += ham(x4);
    }
f2 = 2*f2;
f4 = 4*(f4+ham(x4+h));
tp= h*(f4+f2+ham(a)+ham(b))/6;
```

```
printf("\n Gia tri tich phan la TP= %8.3f",tp);
getch();
return;
} /* Ham can lay tich phan */

float ham(float x)
{
return exp(-x)*sin(x);
}
```

Kết quả của chương trình CT68 là :

Chuong trinh tinh tich phan pp Simpson
Cho can duoi a = 1
Cho can tren b = 5
Cho so diem tinh m = 10
Gia tri tich phan la TP = 0.218

Bài số 69 Tính $\int_a^b f(x)dx$ bằng phương pháp hình thang.

Bài giải Cho hàm y = f(x) liên tục trong khoảng [a,b] tích phân xác định của hàm y = f(x) trong khoảng [a,b] được tính gần đúng bằng tổng các hình thang nguyên tố.

Giả thiết khoảng [a,b] được chia thành m đoạn có bước tích phân $h=(b-a)/m$ thì $x_i = a + ih$, $i = 0,1,...,m$.

Tích phân xác định được tính gần đúng bằng tổng các hình thang :

$$\int_a^b f(x)dx = h\,[\frac{f(a)+f(b)}{2} + \sum_{i=1}^{m-1} f(a+ih)] + R_m(f)$$

trong đó phần dư $R_m(f) = -\dfrac{(b-a)^3}{12m^2} f''(\xi) \quad ,\xi \in [a,b]$

Chương trình CT69 gồm các việc sau :

- Xác định hàm cần lấy tích phân bằng float ham (float x), trong bài này chọn hàm $y = e^{-x}\sin x$.

- Đưa số liệu vào gồm cận dưới a, cận trên b, số điểm tính m.

- Dùng chu trình for để tính tổng giá trị tích phân theo phương pháp hình thang và ghi kết quả.

```
/* CT69 Chuong trinh tinh tich phan pp.hinh thang */
#include <stdio.h>
#include <math.h>
float ham(float);

void main()
{
int i,m;
float a,b,x,s,h,tp;
printf("\n Chuong trinh tinh tich phan pp. hinh thang");
printf("\n Cho can duoi a = ") ;scanf("%f",&a);
printf("\n Cho can tren b = ") ;scanf("%f",&b);
printf("\n Cho so diem tinh m = ") ;scanf("%d",&m);
h = (b-a)/m;
x = a;
s = (ham(a)+ham(b))/2;
for ( i =1; i< m-1;i++)
    {
    x += i*h;
```

```
    s += ham(x);
    }
  tp =s*h;
  printf("\n Gia tri tich phan TP = %f",tp);
  getch();
  return;
  }      /* Ham can lay tich phan */

  float ham(float x)
  {
  return exp(-x)*sin(x);
  }
```

Ta được kết quả chương trình CT69 như sau :

Chuong trinh tinh tich phan pp. hinh thang
Cho can duoi a = 1
Cho can tren b = 5
Cho so diem tinh m = 10
Gia tri tich phan TP = 0.187901

Bài số 70 Tính tích phân kép $\sigma = \int\int f(x,y)dxdy$ bằng phương pháp Gauss - Legendre ở n điểm.

Với n ∈ (3,4,5,6,7,8,9,10,15,20,25,30). Các hoành độ u_1, ..., u_n và các tâm w_1, ...,w_n được tính toán trước.

Trong trường hợp n = 3 ta có :

u[1] = -0.774596   w[1] = 0,555555
u[2]] = 0   w[2] = 0,888888
u[3] = -u[1]   w[3] = w[1]

$$\sigma \approx \frac{h_x h_y}{4} \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \sum_{I=1}^{N} \sum_{J=1}^{N} W_I W_J \; f\left(\frac{h_x}{2} u_I + x_{i-1/2}, \frac{h_y}{2} u_J + y_{j-1/2}\right)$$

Thành phần n_0 và các bước $h_x^{(o)} = b_1 - a_1$, $h_y^{(o)} = b_2 - a_2$ đặt trong các biến l kiểu nguyên và h_x, h_y kiểu thực. Thành phần q_0 kiểu thực.

Tiếp theo ở mỗi bước k ($1 \leq k \leq m$) thành phần l_k và các bước $h_x^{(k)} = (b_1 - a_1)/l_k$, $h_y^{(k)} = (b_2 - a_2)/l_k$ được đặt trong l và h_x, h_y, thành phần q_k đặt vào q_1. Nếu $| q_k - q_{k-1} | < \delta |q_{k-1}|$ nội dung của Q_1 đưa vào biến tích phân tp rồi quá trình ngừng. Nếu không nội dung của Q_1 đưa vào Q_o trước khi chuyển qua lần lặp tiếp theo.

Hàm cần lấy tích phân được định nghĩa trong float hàm (float x, float y) $f(x,y) = (x + y)\sin x \sin y$.

```
/* CT70 Chuong trinh tich phan kep pp.Gauss-Legendre */
#include<stdio.h>
#include<math.h>
float ham(float,float);

void main()
{
int i,i0;j,j0,k,l,m,it,n0;
float t,a1,b1,a2,b2,dta,hx,hy,q0,q1,s,x,x0,y,y0,tp;
float u[15],w[15];
printf("\nChuong trinh tich phan kep pp.Gauss-Legendre");
printf("\n VAO DU LIEU ");
printf("\n Cho can duoi a1 =") ; scanf("%f",&a1);
printf("\n Cho can tren b1 =") ; scanf("%f",&b1);
printf("\n Cho can duoi a2 =") ; scanf("%f",&a2);
printf("\n Cho can tren b2 =") ; scanf("%f",&b2);
printf("\n Do chinh xac dta=") ; scanf("%f",&dta);
```

```
printf("\n So lan lap max m =");scanf("%d",&m);
printf("\n So diem tinh toan no =3"); n0=3;
l=1;hx=b1-a1;hy=b2-a2;s=0;
x0=a1+hx/2;y0=a2+hy/2;
u[1]=-0.774596; w[1]=0.555555;
u[2]=0; w[2]=0.888888;
u[3]=-u[1]; w[3]=w[1];
for(i0=1;i0<=n0;i0++)
    {
    x =hx*u[i0]/2+x0;
    for(j0=1;j0<=n0;j0++)
        {
        y=hy*u[j0]/2+y0;
        s=s+w[i0]*w[j0]*ham(x,y);
        }
    }
q0=hx*hy/4;
it=-1; k=1;
while(it==-1 && k<=m)
    {
    l=2*l;hx=hx/2;hy=hy/2;s=0;
    for(i=1;i<=l;i++)
        {
        x0=a1+(i-0.5)*hx;
        for(j=1 ;j<=l ;j++)
            {
            y0=a2+(j-0.5)*hy;
            for(i0=1 ;i0<=n0 ;i0++)
```

```
            {
            x=hx*u[i0]/2+x0;
            for(j0=1;j0<=n0;j0++)
                {
                y=hy*u[j0]/2+y0;
                s=s+w[i0]*w[j0]*ham(x,y);
                }
            }
        }
    }
  q1=hx*hy*s/4;
  if( abs(q1-q0)<dta*abs(q0))
      {
      tp=q1;it=1;
      }
  else
      {
      q0=q1;k++ ;
      }
  }
printf(" \n Gia tri tich phan la TP  =%f",tp);
getch();
return;
} /* Ham can lay tich phan */
float ham(float x,float y)
{
return (x+y)*sin(x)*sin(y);
{
```

Trong bài này nếu lấy miền $\mathfrak{D} = [\ 0\ ,\ \pi\ ]^2$ và $n_o = 3$ ta được

Chuong trinh tich phan kep pp.Gauss - Legendre

VAO DU LIEU

Cho can duoi a1 = 0

Cho can tren b1 = 3.14159

Cho can duoi a2 = 0

Cho can tren b2 = 3.14159

Do chinh xac dta = 0.001

So lan lap max m = 10

So diem tinh toan no = 3

Gia tri tich phan la TP = 12.566351

Bài số 71 Tính tích phân kép $\sigma = \iint\limits_{\mathfrak{D}} f(x,y)dxdy$ bằng phương pháp Romberg.

Hàm f(x) liên tục trong miền $\mathfrak{D} = [a_1,b_1] \times [a_2,b_2]$. Vì f(x) liên tục trong miền $\mathfrak{D}$ nên có thể viết :

$$\sigma = \int_{a_1}^{b_1} dx \int_{a_2}^{b_2} f(x,y)dy$$

Chia các cận $[a_1,b_1]$ và $[a_2,b_2]$ thành các đoạn nhỏ, nghĩa là chọn các bước tích phân h_x và h_y với các n_x và n_y là các số nguyên :

$$h_x = \frac{b_1 - a_1}{n_x}\ , \qquad h_y = \frac{b_2 - a_2}{n_y}$$

giá trị của x và y tại các điểm lấy tích phân là

$$x_i = a_1 + ih_x\ ,\ i = 1,\dots,n_x$$

$$y_i = a_2 + jh_y\ ,\ j = 1,\dots,n_y$$

Miền $\mathfrak{D}$ được chia thành các miền P_{ij} với $P_{ij} = [x_{i-1},x_i] \times [y_{j-1},y_j]$

$i = 1,\dots,n_x\ ,\quad j = 1,\dots,n_y$

Ta nhận được tích phân kép là một tổng như sau :

$$\sigma = \sum_{i=1}^{n_x}\sum_{j=1}^{n_y} \iint_{P_{ij}} f(x,y)dxdy = \sum_{i=1}^{n_x}\sum_{j=1}^{n_y} \sigma_{ij}$$

Theo công thức điểm giữa ta có :

$$\sigma_{ij} \approx h_x h_y f(x_{i-1/2}, y_{j-1/2})$$

$$\sigma \approx h_x h_y \sum_{i=1}^{n_x}\sum_{j=1}^{n_y} f(x_{i-1/2}, y_{j-1/2})$$

Chương trình tính tích phân kép trong miền **D** bằng phương pháp Romberg gồm các bước sau :

- Hàm cần lấy tích phân f(x,y) được định nghĩa trong float ham (float x, float y).

- Đưa vào các giá trị cận tích phân a_1,b_1 và a_2,b_2 , độ chính xác dta và số lần lặp cực đại m. Ở mỗi giai đoạn k, $0 \le k \le m$, số n_k điểm, bước $h_x^{(k)} = (b_1 - a_1)/n_k$ của đoạn chia nhỏ $[a_1,b_1]$ và bước $h_y^{(k)} = (b_2 - a_2)/n_k$ của đoạn chia nhỏ $[a_2,b_2]$ được sắp xếp trong các biến nguyên l. Khi k = 0 thành phần t1 được đưa vào bảng t.

Nếu $k \geq 1$ bảng t chứa ở các vị trí tương ứng các thành phần $t_i^{(k-i)}$; i=1,...,k. Thành phần $t_{k+1}^{(0)}$ được tính và đặt vào thành phần chỉ số k + 1 của t.

Nếu $| t_{k+1}^{(0)} - t_k^{(0)} | < \delta$ thành phần $t_{k+1}^{(0)}$ được đưa vào giá trị tích phân tp. Khi it = - 1 quá trình ngừng, nếu không thành phần $t_i^{(k-i+1)}$, i = k,...,1 được tính và đặt vào vị trí tương ứng của t. Ở mỗi giai đoạn i, số $2^{2(k-i+1)}$ ở trong C và nếu $| t_i^{(k-i+1)} - t_{i-1}^{(k-i+1)} | < \delta$ $(i \geq 2)$ thành phần $t_i^{(k-i+1)}$ đưa vào tp, it bằng 1 và quá trình ngừng.

```
/* CT71 Chuong trinh tinh tich phan kep pp.Romberg */
#include <stdio.h>
#include <math.h>
float ham( float,float);
```

```
void main()
{
int i,it,j,k,l,m;
float a1,b1,a2,b2,dta,c,hx,hy,s,x,y,tp;
float t[15];
printf("\n Chuong trinh tinh tich phan kep pp. Romberg ");
printf("\n VAO DU LIEU ");
printf("\n Can duoi a1 = "); scanf("%f",&a1);
printf("\n Can tren b1 = "); scanf("%f",&b1);
printf("\n Can duoi a2 = "); scanf("%f",&a2);
printf("\n Can tren b2 = "); scanf("%f",&b2);
printf("\n Do chinh xac dta = "); scanf("%f",&dta);
printf("\n So lan lap max m = "); scanf("%d",&m);
l =1; hx = b1-a1; hy = b2-a2;
t[1] = hx*hy*ham(a1+0.5*hx,a2+0.5*hy);
it= -1;k = 1;
while(it== -1 && k<=m)
    {
    l = 2*l;hx =hx/2;hy =hy/2;
    s = 0;
    for (i=1;i<=l;i++)
        {
        x =a1+(i-0.5)*hx;
        for( j=1; j<l;j++)
            {
            y = a2+(j-0.5)*hy;
            s = s+ham(x,y);
            }
```

```
      }
   t[k+1] =hx*hy*s;
   if (abs(t[k+1]-t[k])<dta)
      {
      tp = t[k+1]; it = 1;
      }
   else
      {
      c =1; i =k;
      while(it == -1 && i>=1)
         {
         c = 4*c ;
         t[i] = (c*t[i+1]-t[i])/(c-1) ;
         if (i>1)
            if( abs(t[i]-t[i-1])<dta)
               { tp =t[i] ;
               it = 1 ;
               }
         i--;
         }
      }
   k++;
   }
printf("\n Gia tri tich phan la TP = %f ",tp);
getch();
return;
}
```

```
/* Ham can lay tich phan */
float ham( float x, float y)
{
return (x+y)*sin(x)*sin(y);
}
```

Kết quả chương trình CT71 như sau :

Chuong trinh tinh tich phan kep pp.Romberg

VAO DU LIEU

Can duoi a1 = 0

Can tren b1 = 3.14159

Can duoi a2 = 0

Can tren b2 = 3.14159

Do chinh xac dta = 0.001

So lan lap max m = 10

Gia tri tich phan la TP = 12.520854

Chương 7

GIẢI PHƯƠNG TRÌNH PHI TUYẾN

Bài số 72 Tìm nghiệm của hàm phi tuyến trong khoảng [a,b] bằng phương pháp chia đôi cung.

Bài giải Đầu tiên sơ bộ xác định khoảng nghiệm một cách hợp lý và tiến hành lặp lại quá trình tính toán sau đây :

- lấy khoảng giữa m = (a+b)/2

- tính toán f(m)

- nếu f(m) = 0, nghiệm là m

- nếu f(a).f(m) < 0 có một nghiệm trong khoảng [a,m], thay thế khoảng [a,b] bằng [a,m] bằng cách đặt b = m

- nếu f(a).f(m) > 0 có một nghiệm trong khoảng [b,m], thay thế [a,b] bằng [b,m] bằng cách đặt a = m.

Khi | b-a | nhỏ hơn độ chính xác mong muốn quá trình tính toán ngừng và nghiệm gần đúng f(m) ≈ 0.

Chương trình CT72 tìm nghiệm của hàm sinx.

```
/* CT72 Chuong trinh tim cuc tri phuong phap chia doi cung */
#include<stdio.h>
#include<math.h>
```

```
int nghiem (double(*(double)(),double,double,double *,double);

void main()
{
double a,b, eps,z;
printf("\n Cho can duoi a = ");scanf("%f",&a);
printf("\n Cho can tren b = ");scanf("%f",&b);
printf("\n Cho do chinh xac eps = ");scanf("%f",&eps);
nghiem ( sin, a,b,&z,eps);
printf("\n Nghiem z = %f",z);
}

int nghiem(double (* f)(double),double a,
                double b,double *zero ,double eps)
{
double m, /* khoang dang xet */
       fm, /* gia tri ham fm */
       fa, /* giatri ham tai a */
       fb; /* gia tri ham tai b */
fa = (*f)(a);
fb = (*f)(b);
if(fa*fb >= 0 || a >= b) return(-1);
while (b-a > eps)
   {
   m = (b+a)/2.0;
   fm = (*f)(m);
   if (fm == 0) break;
   if( fa*fm < 0 )
      { b = m;
```

```
        fb = fm;
        }
    else
        { a = m;
        fa = fm;
        }
    }
 * zero = m;
 return(0);
 }
```

Kết quả ta được :

Cho can duoi a = -1

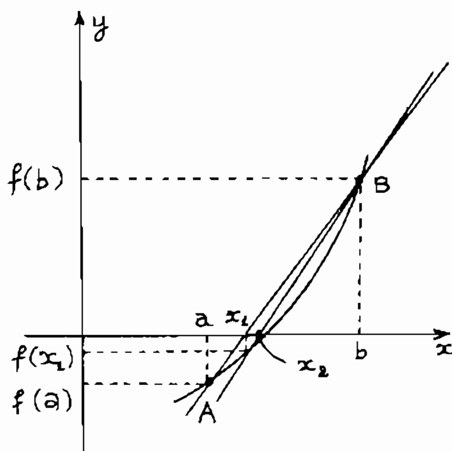
Cho can tren b = 3

Cho do chinh xac eps = 0.01

Nghiem z = 0.000000

Bài số 73 Tìm nghiệm của phương trình phi tuyến y = f(x) bằng phương pháp cát tuyến.

Bài giải Xét đồ thị hàm số y= f(x) hình 7-1. Giả thiết f(a)<0 và f(b)>0. Các điểm của đồ thị 'A[a, f(a)] và B[b, f(b)] được nối với nhau bằng một dây cung. Hoành độ x_1 của giao điểm dây cung AB và trục Ox là giá trị gần đúng của nghiệm tính theo công thức :



Hình 7-1

$$x_1 = a - \frac{(b-a)f(a)}{f(b) - f(a)}$$

trong đó x_1 thuộc khoảng [a,b]. Giả thiết $f(x_1) < 0$ khi đó được khoảng mới hẹp hơn là $[x_1,b]$. Nối các điểm $A_1[x_1,f(x_1)]$ và B[b,f(b)] ta được giao điểm của dây cung với trục ox ở bước thứ hai theo công thức:

$$x_2 = x_1 - \frac{(b-x_1)f(x_1)}{f(b) - f(x_1)}$$

quá trình tính lặp tiếp tục ta được dãy số a, x_1, x_2 ... tiến dần tới nghiệm. Quá trình tính dừng khi sai số $| x_i - x_{i-1} | < \varepsilon$.

```
/* CT73 Chuong trinh giai pt phi tuyen pp. cat tuyen */
#include <stdio.h>
#include <math.h>
#include <conio.h>
#define EPSI 0.005
float Ham(float);

void main()
{
float a,b,fa,fb,x,dx;
clrscr();
printf("\n\t\t\tPHUONG PHAP CAT TUYEN GIAI HAM PHI
TUYEN");
printf("\n\t\t\t\tHAM SO Y = EXP(-x)-x*x*x\n");
printf("\nCho cac can x cua ham\n");
printf("Cho a = ");scanf("%f",&a);
printf("Cho b = ");scanf("%f",&b);
fa = Ham(a);
fb = Ham(b);
```

```
x = a;
dx = fa*(b-a)/(fa-fb);
while (abs(dx)>EPSI)
    {
    x = a+dx;
    b = a;
    a = x;
    fb = fa;
    fa = Ham(x);
    dx = fa*(b-a)/(fa-fb);
}
printf("\n Nghiem x = %f",x);
printf("\n Gia tri ham = %f",Ham(x));
getch();
return;
}

float Ham(float x)
{
return(exp(x)+x*x*x);
}
```

Chương trình CT73 cho ta kết quả sau :

Cho cac can x cua ham

Cho a = - 3

Cho b = 3

Nghiem x = -0.815903

Gia tri ham = -0.100904

Bài số 74 Tìm nghiệm của hàm phi tuyến $y = f(x)$ bằng phương pháp lặp Newton.

Bài giải Để tìm nghiệm gần đúng của hàm $y = f(x)$ ta khai triển hàm thành chuỗi Taylor :

$$f(x) = f(x_n) + (\alpha - x_n).f'(x_n) + \dots$$

Vì $f(\alpha) = 0$ suy ra :

$$\alpha = x_n - \frac{f(x_n)}{f'(x_n)} \text{ với } f'(x) \neq 0$$

Phương pháp lặp Newton tiến hành theo thuật toán :

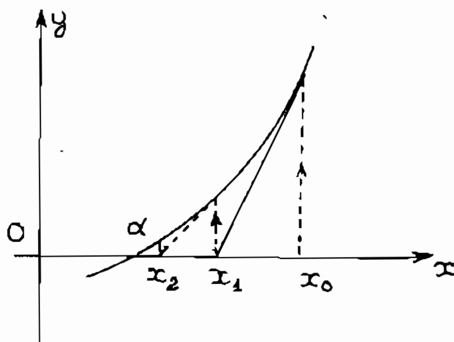
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \; n=0,1\dots$$

Về mặt hình học phương pháp lặp Newton (còn gọi là phương pháp tiếp tuyến) tiến hành tìm nghiệm gần đúng α với hoành độ x_α là điểm cắt của tiếp tuyến đường cong tại điểm x_{k-1}, đường cong được thay bằng đường tiếp tuyến (hình 7-2).

Phương pháp lặp sẽ hội tụ nếu f(x) liên tục và có đạo hàm ở lân cận nghiệm α. Phép lặp sẽ dừng khi sai số $|x_{n+1} - x_n| \leq \varepsilon$.

Phương pháp lặp Newton hội tụ nhanh được coi là phương pháp cơ bản để tìm nghiệm của hàm phi tuyến.

Chương trình CT74 gồm việc xác định hàm f(x) và đạo hàm cấp một f'(x) và thực hiện một chu trình do while để tính lặp. Nếu bước lặp lớn hơn 100 sẽ báo bài toán không hội tụ.



Hình 7-2

```
/* CT74 Chuong trinh giai pt phi tuyen pp.Newton */

#include<stdio.h>
#include<math.h>
float Ham(float);
float Dham(float);

void main()
{
int i;
float x0,epsi,t;
float x[100];
printf("\n GIAI PT PHI TUYEN BANG PP. NEWTON ");
printf("\n Cho gia tri dau x0 =") ;scanf("%f",&x0);
printf("\n Cho sai so epsilon =") ;scanf("%f",&epsi);
i=1;
x[i]= x0;
do
    {
    x[i+1] =x[i]-Ham(x[i])/Dham(x[i]);
    t = abs(x[i+1]-x[i]);
    x[i] = x[i+1]
    i++;
    if (i>100) printf("\n Bai toan khong hoi tu ");
    }
while( abs(t) >epsi);
printf("\n Nghiem gan dung x =%f",x[i]);
printf("\n Gia tri ham = %f",Ham(x[i]));
```

```
getch();
return;
}

/* Ham can tim nghiem */
float Ham(float x)
{
return(exp(-x)-x*x);
}

float Dham(float);
/* Dao ham */
float Dham(float x)
{
return(-exp(-x)-2*x);
}
```

Ta được kết quả :

Cho sai so epsilon = 0.001
Nghiem gan dung x = 0.733044
Gia tri ham = -0.056908

Chương 8

TỐI ƯU HÓA

Chương này trình bày một số thuật toán và chương trình tối ưu hóa các hàm một biến và nhiều biến. Đối với hàm một biến số thực đưa ra thuật toán và chương trình tìm cực trị theo phương pháp tiết diện vàng và phương pháp gradient. Đối với hàm nhiều biến trình bày thuật toán và chương trình tìm cực trị hàm nhiều biến bằng phương pháp gradient.

Nếu hàm f(x) xác định trong miền D thuộc không gian thực, hàm f có cực tiểu tổng thể tại điểm x_0 thuộc miền D nếu :

$$(\forall x \in D) \qquad f(x) \geq f(x_0)$$

Hàm f có cực đại tổng thể tại điểm x_0 thuộc miền D nếu :

$$(\forall x \in D) \qquad f(x) \leq f(x_0)$$

f có cực tiểu cục bộ ở điểm x_0 nếu tồn tại một điểm lân cận Y của x_0 sao cho :

$$(\forall x \in V \cap D) \quad f(x) \geq f(x_0)$$

và cực đại cục bộ tại x_0 sao cho :

$$(\forall x \in V\ D) \quad f(x) \leq f(x_0)$$

Nếu hàm f(x) có đạo hàm thì một số cực trị của nó có thể được xác định bằng cách tìm nghiệm phương trình $f'(x) = 0$. Tuy nhiên có những điểm tại đó đạo hàm bậc nhất bằng không không phải là điểm cực trị (điểm uốn) vì thế các kỹ thuật tìm kiếm cực trị của hàm số nói chung không dựa trên việc tìm nghiệm của phương trình $f'(x) = 0$.

Bài số 75 Tìm cực trị của hàm một biến F(x) trong khoảng [a, b] bằng phương pháp tiết diện vàng.

Bài giải Chuỗi q_k được coi là hội tụ về q, ta có :

$$\lim_{k \to +\infty} q_{k-1} = \lim_{k \to +\infty} q_k = q$$

và $q_k = 1 + \dfrac{1}{q_{k-1}}$, do đó ta có $q = 1 + \dfrac{1}{q}$, suy ra $q^2 - q - 1 = 0$ và có nghiệm dương là $\dfrac{\sqrt{5}+1}{2} \approx 1{,}6180339887...$ gọi là số vàng.

suy ra $r = \dfrac{1}{q} = q - 1 = \dfrac{\sqrt{5}-1}{2} \approx 0{,}6180339887...$

Phương pháp tiết diện vàng giữ tỷ số $\dfrac{l_{k+1}}{l_k}$ của hai khoảng liên tiếp không đổi bằng cách lấy $\dfrac{l_{k+1}}{l_k} = r$, giữ nguyên quan hệ $l_k = l_{k+1} + l_{k+2}$, $k = 1, 2, ...$

Ta có các biểu thức $l_{k+1} = rl_k$, $l_{k+1} = r^k l_1$, $l_k = ql_{k+1}$, $k = 1, 2, ...$

Thuật toán tiến hành như sau :

Đặt

$$\left[\begin{array}{l} \left[\begin{array}{l} r = \dfrac{\sqrt{5}-1}{2} \\ a_1 = a\ ,\ b_1 = b\ ,\ l_1 = b_1 - a_1 \\ l_2 = rl_1 \\ \alpha_1 = b_1 - l_2\ ,\ \beta_1 = a_1 + l_2 \\ Y_{\alpha 1} = f(\alpha_1)\ ,\ Y_{\beta 1} = f(\beta_1) \end{array}\right. \\ \left[\begin{array}{lll} l_{k+2} = rl_{k+1} & & \\ \text{nếu } Y_{\beta k} < Y_{\alpha k}\ , & a_{k+1} = \alpha_k\ ,\ b_{k+1} = b_k & \\ & \alpha_{k+1} = \beta_k\ ,\ Y_{\alpha k+1} = Y_{\beta k} & \\ & \beta_{k+1} = a_{k+1} + l_{k+2}\ ,\ Y_{\beta k+1} = f(\beta_{k+1}) & k=1,2,.. \\ \text{nếu không}\ , & a_{k+1} = a_k\ ,\ b_{k+1} = \beta_k & \\ & \beta_{k+1} = \alpha_k\ ,\ Y_{\beta k+1} = Y_{\alpha k} & \\ & \alpha_{k+1} = b_{k+1} - l_{k+2}\ ,\ Y_{\alpha k+1} = f(\alpha_{k+1}) & \end{array}\right. \end{array}\right.$$

Ta cố định độ chính xác $\delta > 2\eta$ và ngừng quá trình khi $l_{k+2} \leq \delta$;

ta có $\begin{cases} a_{k+2} = \alpha_{k+1}, b_{k+2} = b_{k+1} \text{ nếu } Y_{\beta k+1} < Y_{\alpha k+1} \\ a_{k+2} = a_{k+1}, b_{k+2} = \beta_{k+1} \text{ nếu không} \end{cases}$

do đó ta có :

$$x_{min} \approx (a_{k+2} + b_{k+2}) / 2 \text{ và } y_{min} = f(\ (a_{k+2} + b_{k+2}) /2\ )$$

Chương trình gồm xác định hàm $f(x) = (4x^2 + 3x)e^{-2x}$ và đưa vào các cận :

a : cận dưới của khoảng tìm kiếm

b : cận trên của khoảng tìm kiếm, dta là sai số

```
/* CT75 Chuong trinh cuc tri theo tiet dien vang */
#include <stdio.h>
#include <math.h>
#include <conio.h>
float Ham(float );

void main ()
{
float a,b,dta;
float xmin,ymin;
float l,r,to;
float yalpha,ybeta,alpha,beta;
clrscr();
printf("\nTim cuc tri ham F(x) p p TIET DIEN VANG\n");
printf("\nCho can duoi a = ");scanf("%f",&a);
printf("\nCho can tren b = ");scanf("%f",&b);
printf("\nCho sai so dta = ");scanf("%f",&dta);
r = (sqrt(5)-1)/2;
```

```
l = b-a;
l = r*l;
alpha = b-l;
beta = a+l;
yalpha = Ham(alpha);
ybeta = Ham(beta);
do
    {
    l = r*l;
    if (ybeta<yalpha)
        {
        a = alpha;
        alpha = beta;
        yalpha = ybeta;
        beta = a+l;
        ybeta = Ham(beta);
        }
    else
        {
        b = beta;
        beta = alpha;
        ybeta = yalpha;
        alpha = b-l;
        yalpha = Ham(alpha);
        }
    if (alpha>beta)
        {
        to = alpha;
```

```
        alpha = beta;
        rbeta = to;
        to = yalpha;
        yalpha = ybeta;
        ybeta = to;
        }
    }
while (l<=dta);
if (ybeta<yalpha) a = alpha;
else b = beta;
xmin = (a+b)/2;
ymin = Ham(xmin);
printf("\nKET QUA\t Xmin = %f\t\tYmin = %f",xmin,ymin);
getch();
return;
}

/* Ham can tim cuc tri */
float Ham(float x)
{
return (4*x*x+3*x)*exp(-2*x) ;
}
```

Ta được kết quả chương trình CT75 :

Tim cuc tri ham F(x) p p TIET DIEN VANG

Cho can duoi a = 0

Cho can tren b = 3

Cho sai so dta = 0.001

KET QUA Xmin = 2.427051 Ymin = 0.240466

Bài số 76 Tìm cực trị của hàm một biến f(x) trong khoảng [a, b] bằng phương pháp gradient.

Bài giải

Việc tìm cực trị của hàm một biến f(x) dẫn tới tìm nghiệm của phương trình $f'(x) = 0$

Gọi h là bước tính $h = \dfrac{(b-a)}{n}$, n là số điểm tính toán ta có :

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

$$f''(x) \approx \frac{f(x+h) + f(x-h) - 2f(x)}{h^2}$$

Quá trình tính lặp theo phương pháp gradient cho theo công thức

$$x_{i+1} = x_i - h\,\frac{f(x+h) - f(x-h)}{2[f(x+h) + f(x-h) - 2f(x)]}$$

Chương trình tìm cực trị của hàm một biến theo phương pháp gradient gồm các bước sau :

- Đọc theo giá trị cận dưới a, bước h và sai số cho phép epsilon.
- Xây dựng hàm cần tìm cực trị, trong trường hợp bài toán có

$$f(x) = e^{-x}\sin x$$

- Tổ chức một chu trình do ... while để tính lặp. Chương trình dừng khi $|x_{i+1} - x_i| \le$ epsilon.

```
/* CT76 Chuong trinh cuc tri ham 1 bien */
#include <stdio.h>
#include <math.h>
float Ham(float );
```

```
void main ()
{
float a,h,eps,x,xe,ye,y1,y2,y3;
clrscr();
printf("\nTim cuc tri ham mot bien ");
printf("\nCho can duoi a = ");scanf("%f",&a);
printf("\nCho sai so eps ="); scanf("%f",&eps);
printf("\nCho buoc h = ");scanf("%f",&h)
do
    {
    x = a;
    y1 = Ham(x-h);
    y2 = Ham(x);
    y3 = Ham(x+h);
    a = x-h*(y3-y1)/(2*(y3+y1-2*y2));
    } while (abs(x-a) > eps);
xe = a;
ye = Ham(xe);
printf("\n Cuc tri tai xe = %f ,ye = %f",xe,ye);
getch();
return;
}

/* Ham can tim cuc tri */
float Ham(float x)
{
return exp(-x)*sin(x) ;
}
```

Kết quả chương trình CT76 như sau :

Tim cuc tri ham mot bien

Cho can duoi a = 0

Cho sai so eps = 0.001

Cho buoc h = 0.001

Cuc tri tai xe = 0.499996, ye =0.290785

Bài số 77 Tìm cực trị của hàm hai biến f(x, y) bằng phương pháp gradient.

Bài giải Cho hàm hai biến f(x, y), việc tìm cực trị của hàm f(x, y) dẫn đến việc giải hệ phương trình:

$$f'_x(x,y) = 0 \text{ và } f'_y(x,y) = 0$$

Nếu f(x, y) đã cho thì ta tính các đạo hàm riêng và bài toán tìm cực trị dẫn tới tìm nghiệm của hệ phương trình. Việc tính các đạo hàm cấp 2 thường dài, tuy nhiên ta có thể dùng một chu trình lặp theo phương pháp gradient như sau :

Trên mặt f(x,y) ta tìm một cực tiểu theo hướng có độ dốc lớn nhất, nghĩa là theo hướng ∇f. Nếu f(x,y) có cực tiểu tại x_m, y_m ta có quá trình lặp :

$$x_{i+1} = x_i - u_i f'_x(x_i,y_i)$$

$$y_{i+1} = y_i - u_i f'_y(x_i,y_i)$$

u_i có thể xác định ở mỗi bước tính.

Nếu chọn h là bước lặp các gradient theo phương pháp x và y được xác định như sau :

$$gradx = f(x+h, y) - f(x-h, y)$$

$$grady = f(x, y+h) - f(x, y-h)$$

$$grad = (gradx^2 + grady^2)^{1/2}$$

Quá trình tính lặp theo thuật toán sau :

$$x_{i+1} = x_i - l.gradx / grad$$

$$y_{i+1} = y_i - l.grady / grad$$

trong đó l là khoảng cách cho trước.

Chương trình tìm cực trị của hàm hai biến theo phương pháp gradient gồm các bước sau :

- Định nghĩa hàm cần tìm cực trị, ở đây f = (x+y)sinx.siny
- Cho các cận dưới x1, y1, bước h, khoảng cách l, l_{max}
- Tổ chức chu trình tính lặp theo phương pháp gradient.

```
/* CT77 Chuong trinh tinh cuc tri pp.gradient */

#include<stdio.h>
#include<math.h>
float Ham(float,float);

void main()
{
int i,OK = 0;
float l,lmax,h,xl,yl,xm,ym,gradx,grady,grad;
printf("\n Tinh cuc tri ham 2 bien pp. gradient ");
printf("\n Cho can xl = "); scanf("%f",&xl);
printf("\n Cho can yl = "); scanf("%f",&yl);
printf("\n Cho buoc h = "); scanf("%f",&h);
printf("\n Cho khoang cach l= ");scanf("%f",&l);
printf("\n Cho lmax = "); scanf("%f",&lmax);
xm = xl,ym = yl;
```

```
for(i=0;!OK;i++)
  {
  if(i> 1000)
    { OK = 1;
    printf("\n Tinh toan qua dai !\n ");
    }
  gradx = Ham(xm+h,ym)-Ham(xm-h,ym);
  grady = Ham(xm,ym+h)-Ham(xm,ym-h);
  grad = sqrt(gradx*gradx+grady*grady);
  xl = xm, yl = ym;
  l *= 2;
  if( grad!= 0)
    {
    do
      {
      l /= 2;
      xm = xl-l*gradx/grad;
      ym = yl-l*grady/grad;
      if(l<lmax) OK = 1;
      }
      while(Ham(xm,ym) < Ham(xl,yl));
    }
  else OK = 1;
  }
printf("\n Cuc tri tai xm =%f ym =%f",xm,ym);
getch();
return;
}
```

```
/* Ham can tim cuc tri */
float Ham(float x, float y)
{
return(x+y)*sin(x)*sin(y);
}
```

Kết quả chương trình CT77 như sau :

Tinh cuc tri ham 2 bien pp. gradient
Cho can xl = -2
Cho can yl = 2
Cho buoc h = 0.001
Cho khoang cach l = 10
Cho lmax = 15
Cuc tri tai xm = -0.232233 ym = 3.767767

Chương 9

PHƯƠNG TRÌNH VI PHÂN

Trong chương này trình bày một số thuật toán và chương trình giải gần đúng phương trình vi phân cấp một và hệ n phương trình vi phân cấp một bài toán sơ kiện (Bài toán Cauchy).

Cho phương trình vi phân (PTVP) cấp một dạng :

$$y' = f(x,y) \qquad (1)$$

Ta gọi nghiệm của phương trình (1) trong khoảng I là hàm y có đạo hàm trong khoảng I sao cho $\forall x \in I$ có $y'(x) = f(x, y(x))$.

Việc giải phương trình (1) là tìm các nghiệm theo sơ kiện sao cho phương trình vi phân có một giá trị cố định tại một điểm cho trước.

$$\begin{cases} y' = f(x,y) \\ y(a) = y_a \end{cases}, \forall x \in [a,b] \qquad (2)$$

Nếu f liên tục trong khoảng [a, b], thỏa mãn điều kiện Lipschitz :

$$\forall x \in [a, b] \ \forall u,v \in R \text{ thì } |f(x,u) - f(x,v)| \leq L|u-v|, \ L>0$$

phương trình vi phân (2) có một nghiệm duy nhất là y.

Nghiệm bằng số của PTVP (2) là dãy số $y_1, y_2, \dots, y_N$ là các giá trị gần đúng của các giá trị chính xác $y(x_1), y(x_2), \dots, y(x_N)$ của nghiệm chính xác y ở N điểm $x_1<x_2<\dots<x_N$ của khoảng [a,b] với bước $h_i=x_{i+1}-x_i$, $i = 0, \dots, N-1$ với $x_0 = a$ có thể là hằng số hoặc không.

Các phương pháp giải gần đúng có bước không đôi được đặc trưng bằng việc để tính toán một giá trị mới y_{i+1} chỉ cần thông tin tại một điểm (x_i, y_i).

Các phương pháp có bước thay đổi được đặc trưng bằng việc tính toán ở giá trị mới y_{i+1} cần thông tin ở nhiều bước trước và một quá trình ngoại suy.

Bài số 78 Giải phương trình vi phân cấp một $y' = f(x,y)$ thỏa mãn sơ kiện $y(x_0) = y_0$ bằng phương pháp Euler.

Bài giải

Phương pháp Euler là phương pháp gần đúng có bước h không đổi theo công thức lặp :

$$y_{i+1} = y_i + hf(x_i,y_i)\ ,\ i = 0, \ldots , N-1$$

Sai số của phương pháp cỡ h^2.

Để tổ chức chương trình giải PTVP cấp một theo phương pháp Euler CT78 gồm các bước sau :

- Xác định hàm $y' = f(x,y)$
- Cho khoảng tích phân [a,b] và sơ kiện x_0, y_0
- Tổ chức tính lặp theo công thức Euler và ghi kết quả các giá trị $y_i(x_i)$.

```
/* CT78 Chuong trinh giai PTVP pp. EULER */
#include<stdio.h>
#include<math.h>
float ham(float);
void main()
{
int i,m;
```

```
float a,b,h,x0,y0,c1,c2;
float x[100],y[100];
printf("\n Giai PTVP pp.Euler ");
printf("\n Vao du lieu");
printf("\n Cho gia tri dau a =");scanf("%f",&a);
printf("\n Cho gia tri cuoi b=");scanf("%f",&b);
printf("\n Cho so buoc m =");scanf("%d",&m);
printf("\n Cho so kien x0 =");scanf("%f",&x0);
printf("\n Cho so kien y0 =");scanf("%f",&y0);
h=(b-a)/m;
x[0] = x0;
y[0] = y0;
printf("\n Bang ket qua ");
for( i=0; i<=m;i++)
    {
    x[i+1] = x[i]+h;
    y[i+1] = y[i]+h*ham(x[i]);
    printf("\n %-10.3f %-10.3f",x[i],y[i]);
    }
getch();
return;
}

/* Ham can giai */
float ham(float x)
{
return exp(-x)*sin(x);
}
```

Ta được bảng kết quả sau đây :

Giai PTVP pp.Euler

Vao du lieu

Cho gia tri dau a = 0

Cho gia tri cuoi b = 2

Cho so buoc m = 10

Cho so kien x0 = 0

Cho so kien y0 = 0

Bang ket qua

0.000	*0.000*
0.200	*0.000*
0.400	*0.033*
0.600	*0.085*
0.800	*0.147*
1.000	*0.211*
1.200	*0.273*
1.400	*0.329*
1.600	*0.378*
1.800	*0.418*
2.000	*0.450*

Bài số 79 Giải phương trình vi phân cấp một y' =f(x,y) thỏa mãn sơ kiện y' $(x_0) = y_0$ bằng phương pháp Euler cải tiến.

Bài giải

Phương pháp Euler cải tiến là phương pháp gần đúng có bước h không đổi theo công thức lặp sau đây :

$$\left| \begin{aligned} C_1 &= hf(x_i, y_i) \\ C_2 &= hf(x_i + h\ ,\ y_i + C_1) \\ y_{i+1} &= y_i + \frac{1}{2}(C_1 + C_2) \end{aligned} \right| \quad i = 0, \ldots, N\text{-}1$$

sai số của phương pháp cỡ h^3.

Chương trình CT79 gồm các bước sau đây :

- Xác định hàm y' = f(x,y)
- Cho khoảng tích phân [a, b] và sơ kiện $y'(x_0) = y_0$
- Tổ chức tính lặp theo phương pháp Euler cải tiến.

```
/* CT79 Chuong trinh giai PTVP pp. Euler cai tien */

#include<stdio.h>
#include<math.h>
float ham(float);

void main()
{
int i,m;
float a,b,x0,y0,h,c1,c2;
float x[100],y[100];
printf("\n Vao du lieu");
printf("\n Gia tri dau a = ");scanf("%f",&a);
printf("\n Gia tri cuoi b = ");scanf("%f",&b) ;
printf("\n So buoc m = "); scanf("%d",&m);
printf("\n So kien xo ="); scanf("%f",&x0);
printf("\n So kien y0 = "); scanf("%f",&y0);
h = (b-a)/m;
```

```
x[0] = x0;
y[0] = y0;
printf("\n Bang ket qua");
for( i=0; i<=m;i++)
    {
    x[i+1] = x[i]+h;
    c1 = h*ham(x[i]);
    c2 = h* ham(x[i]+h);
    y[i+1] =y[i]+(c1+c2)/2;
    printf("\n %-10.3f %-10.3f",x[i],y[i]);
    }
getch();
return;
}

float ham(float x)
{
return exp(-x)*sin(x);
}
```

Kết quả ta được bảng bảng giá trị y_i (x_i).

Vao du lieu

Gia tri dau a = 0

Gia tri cuoi b = 2

So buoc m = 10

So kien x0 = 0

So kien y0 = 0

Bang ket qua

0.000	*0.000*
0.200	*0.016*
0.400	*0.059*
0.600	*0.116*
0.800	*0.179*
1.000	*0.242*
1.200	*0.301*
1.400	*0.354*
1.600	*0.398*
1.800	*0.434*
2.000	*0.463*

Bài số 80 Giải phương trình vi phân cấp một $y' = f(x,y)$ thoả mãn sơ kiện $y'(x_0) = y_0$ bằng phương pháp Runge - Kutta.

Bài giải Nghiệm gần đúng của PTVP cấp một trong khoảng [a,b], bước $h=x_{i+1} - x_i$ thỏa mãn sơ kiện $y'(x_0) = y_0$ được tìm theo công thức Runge - Kutta cấp 4 :

$$\left[\begin{array}{l} C_1 = hf(x_i,y_i) \\ C_2 = hf\left(x_i+\frac{1}{2} h, y_i+\frac{1}{2} C_1 \right) \\ C_3 = hf\left(x_i+\frac{1}{2} h, y_i+\frac{1}{2} C_3 \right) \\ C_4 = hf(x_i+h, y_i+C_2) \\ y_{i+1} = y_i + \frac{1}{6} (C_1 + 2C_2 + 2C_3 + C_4) \end{array}\right. \qquad i = 0, 1, \dots, N-1$$

Sai số của phương pháp cỡ h^5.

Chương trình CT80 gồm các bước sau :

- Xác định hàm y' = f(x,y)
- Cho khoảng tích phân [a,b] và sơ kiện x_0, y_0
- Tính lặp theo công thức Runge - Kutta cấp 4.

Ta nhận thấy sai số của phương pháp giảm đi khi số điểm tính N tăng lên, nghĩa là bước h nhỏ đi. Tuy nhiên các phương pháp có bước không đổi như Euler, Euler cải tiến và Runge - Kutta có nhược điểm là sai số gặp phải ngoài sai số phương pháp còn phải kể tới sai số làm tròn của máy tính, khi số điểm tính N tăng sai số phương pháp giảm đi nhưng sai số làm tròn tăng lên. Do đó đối với mỗi phương pháp tính có một giá trị N tối ưu không được vượt quá. Giá trị N phụ thuộc vào hàm f và vào máy tính.

```
/* CT80 Chuong trinh giai PTVP pp.Runge-Kutta */

#include<stdio.h>
#include<math.h>
float ham(float);

void main()
{
int i,m;
float a,b,x0,y0,h,k1,k2,k3,k4;
float x[100],y[100];
printf("\n Phuong phap RK 4");
printf("\n Vao du lieu ");
printf("\n Gia tri dau a = ");scanf("%f",&a);
printf("\n Gia tri cuoi b= ");scanf("%f",&b);
printf("\n So buoc tinh m= ");scanf("%d",&m);
```

```
printf("\n So kien x0 = ");scanf("%f",&x0);
printf("\n So kien y0 = ");scanf("%f",&y0);
h = (b-a)/m;
x[0] =x0;
y[0] =y0;
printf("\n Bang ket qua ");
for (i=0;i<=m;i++)
    {
    x[i+1] = x[i]+h;
    k1 = h*ham(x[i]);
    k2 = h*ham(x[i]+h/2);
    k3 = h*ham(x[i]+h/2);
    k4 = h*ham(x[i]+h);
    y[i+1] = y[i]+(k1+2*k2+2*k3+k4)/6;
    printf("\n %-10.3f %-10.3f",x[i],y[i]);
    }
getch();
return;
}

float ham(float x)
{
return exp(-x)*sin(x) ;
}
```

Ta được kết quả chương trình CT80 :

Phuong phap RK = 4

Vao du lieu

Gia tri dau a = 0

Gia tri cuoi b = 2

So buoc tinh m = 10

So kien x0 = 0

So kien y0 = 0

Bang ket qua

0.000	*0.000*
0.200	*0.017*
0.400	*0.061*
0.600	*0.119*
0.800	*0.182*
1.000	*0.246*
1.200	*0.305*
1.400	*0.358*
1.600	*0.402*
1.800	*0.438*
2.000	*0.467*

Bài số 81 Giải hệ phương trình vi phân cấp n thỏa mãn sơ kiện $y(0)=y_0$

Bài giải Phương trình vi phân (PTVP) cấp n có thể đưa về hệ n PTVP cấp một dạng :

$$\begin{cases} y'_1 = g_1(x, y_1, \ldots, y_n) \\ y'_2 = g_2(x, y_1, \ldots, y_n) \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ y'_n = g_n(x, y_1, \ldots, y_n) \end{cases} \qquad (1)$$

trong đó : $g_1, \ldots, g_n$ là các hàm của $x, y_1, y_2, \ldots, y_n$

$y_1, \ldots, y_n$ là các hàm biến số thực của x

y'1,...,y'n là các đạo hàm cấp một của các hàm $y_1,\ldots,y_n$.

Ta gọi nghiệm của hệ PTVP cấp một trong khoảng I của không gian thực R là các hàm $y = (y_1, ..., y_n)$ có đạo hàm trong khoảng I sao cho với mọi x thuộc I ta có :

$$y'(x) = g(x, y(x))$$

Giải hệ PTVP (1) là tìm toàn bộ các nghiệm này. Giải hệ PTVP (1) theo các điều kiện đầu :

$$\begin{cases} y' = g(x,y) \\ y(a) = Y_a \end{cases} \quad x \in [a,b] \qquad (2)$$

nếu g liên tục trong khoảng [a, b] và thỏa mãn điều kiện Lipschitz hệ (2) có nghiệm duy nhất. Nghiệm số của hệ (2) là một dãy các vectơ $Y_1, ..., Y_n$ làm gần đúng các giá trị $y(x_1), ..., y(x_N)$ của nghiệm chính xác y ở N điểm $x_1 < x_2 < ... < x_N$ của khoảng [a,b] với bước $h_i = x_{i+1} - x_i$, $i = 0, ..., N-1$ và $x_0 = a$.

Ta dùng thuật toán tính lặp theo Runge - Kutta cấp 4 như sau :

$$\left| \begin{aligned} C_1 &= hg(x_i, Y_i) \\ C_2 &= hg(x_i + \frac{1}{2}h, Y_i + \frac{1}{2}C_1) \\ C_3 &= hg(x_i + \frac{1}{2}h, Y_i + \frac{1}{2}C_2) \\ C_4 &= hg(x_i + h, Y_i + C_3) \\ Y_{i+1} &= Y_i + \frac{1}{6}(C_1 + 2C_2 + 2C_3 + C_4) \end{aligned} \right| \quad i = 0, 1, ..., N-1$$

Trong chương trình CT81 hệ PTVP cấp n được xác định bằng hàm G có dạng :

```
void G ( int n, double x, double * y, double * w )
```

và được minh họa bằng hệ 2 PTVP cấp một :

$$y'_1 = 1 - 2y_1/x$$

$$y'_2 = y_1 - y_2 - 1 + 2y_1/x$$

với $x \in [1,2]$ thỏa mãn sơ kiện :

$$y_1(x_o) = y_1(1) = 1/3 \text{ và}$$

$$y_2(x_o) = y_2(1) = -1/3$$

Chương trình CT81 gồm các bước :

- Xác định hàm G là hệ n PTVP cấp một cần giải.
- Cho sơ kiện bài toán.
- Cho bước tích phân h và khoảng nghiệm [a,b]
- Dùng chu trình for để tìm nghiệm gần đúng của hệ PTVP (2) theo phương pháp Runge - Kutta cấp 4 và in các kết quả :

$$y_1(x_1), \ldots, y_1(x_n)$$

$$y_2(x_1), \ldots, y_2(x_n)$$

....

$$y_n(x_1), \ldots, y_n(x_n)$$

```
/* CT81 Chuong trinh giai he PTVP cap n pp Runge-Kutta 4 */
#include<stdio.h>
#include<math.h>
#define N 21
typedef double vec[N];
vec c1,c2,c3,c4,y,v,w;
void G(int,float,double*,double*);

void main()
{
int i,k,n,l;
float x,x0,h,t;
printf("\n Cho so phuong trinh n = ");scanf("%d",&n);
```

```
printf("\n Cho buoc h = "); scanf("%f",&h);
printf("\n Cho so kien x0 = ");scanf("%f",&x0);
/* Phuong phap Runge-Kutta cap 4 */
for(l=1;l<=n;l++)
    { printf("\n Cho so kien y[%d]= ",l);
    scanf("%f",&t);y[l]= t;
    }
x = x0;
for( i=1;i<=100;i++)
    {G(n,x,y,w);
    for(k=1;k<=n;k++)
        {
        c1[k] = h*w[k];
        v[k] = y[k]+c1[k]/2;
        }
    G(n,x+h/2,v,w);
    for(k=1;k<=n;k++)
        {
        c2[k] = h*w[k];
        v[k] = y[k]+c2[k]/2;
        }
    G(n,x+h/2,v,w);
    for(k=1;k<=n;k++)
        {
        c3[k] = h*w[k];
        v[k] = y[k]+c3[k];
        }
    G(n,x+h,v,w);
```

```
    for(k=1;k<=n;k++)
        {
        c4[k] = h*w[k];
        y[k]=y[k]+(c1[k]+2*c2[k]+2*c3[k]+c4[k])/6;
        printf("\n x = %f y[%d] = %f\n",x,k,y[k]);
        }
    x += h;
    }
getch();
return;
}

void G( int n,float x,double *y,double *w) /* PTVP dang chuan */
{
w[1] = 1-2*y[1]/x;
w[2] = y[1]+y[2]-1+2*y[1]/x;
}
```

Ta được kết quả chương trình như sau :

Cho so phuong trinh n = 2

Cho buoc h = 0.01

Cho so kien x0 = 1

Cho so kien y1 = 0.333

Cho so kien y2 = -0.333

x = 1.000000	*y1 = 0.333333*	*y2 = -0.333333*
x = 1.100000	*y1 = 0.366666*	*y2 = -0.366666*
x = 1.300000	*y1 = 0.433333*	*y2 = -0.433333*
x = 1.700000	*y1 = 0.566666*	*y2 = -0.566666*
x = 2.000000	*y1 = 0.666666*	*y2 = -0.666666*

Chương 10

XÁC SUẤT THỐNG KÊ

Bài số 82 Tìm phân bố nhị thức biết số phép thử n = 10, xác suất **xuất** hiện biến cố ngẫu nhiên trong mỗi phép thử p = 0,45 , x = 4.

Bài giải Biến ngẫu nhiên x có thể nhận các giá trị nguyên 0, 1, 2, ... n. Xác suất để trong n phép thử biến cố xuất hiện x lần được tính theo công thức Bernouille :

$$P_{x,n} = C_n^x \, p^x \, q^{n-x} \text{ , trong đó } q = 1\text{-}p$$

Tính tổ hợp chập x trong n phần tử được tính theo công thức :

$$C_n^x = \frac{n!}{x!(n-x)!}$$

do đó ta được :

$$P_{x,n} = \frac{n!}{x!(n-x)!} p^x.(1-q)^{n-x}$$

Chương trình phân bố nhị thức gồm các bước sau :

- Đọc số lần thử n = 10, xác suất p = 0,45 , biến cố xuất hiện x=4 sau đó gọi hàm tính giai thừa.

- Đặt $s_1 = n!$, $s_2 = x!$, $x_3 = (n-x)!$
- Dùng hàm pow để tính các hàm mũ : $s_4 = pow(p,x)$
$s_5 = pow(1-p, n-x)$

Kết quả ta được phân bố nhị thức :

$$B = \frac{s_1}{s_2.s_3} . s_4 . s_5$$

```
/* CT82 Chuong trinh tinh phan bo nhi thuc */
#include<stdio.h>
#include<math.h>
float GIAITHUA(int);

void main()
{
int n,x;
float p,s1,s2,s3,s4,s5,nt;
printf("\n So phep thu n = "); scanf("%d",&n);
printf("\n So lan xuat hien bien co x = ");scanf("%d",&x);
printf("\n Xac suat xuat hien x p = ");scanf("%f",&p);
s1 = GIAITHUA(n);
s2 = GIAITHUA(x);
s3 = GIAITHUA(n-x);
s4 = pow(p,x);
s5 = pow(1-p,n-x);
nt = s1*s4*s5/(s2*s3);
printf("\n Phan bo nhi thuc : %f ",nt);
getch();
return;
}
```

```
/* Tinh giai thua */
float GIAITHUA(int n)
{
if(n==1 || n==0)
    return(1);
else
    return(n*GIAITHUA(n-1));
}
```

Kết quả chương trình CT82 như sau :

So phep thu n = 3
So lan xuat hien bien co x = 4
Xac suat xuat hien x p = 0.45
Phan bo nhi thuc : 0.238367

Bài số 83 Dùng phân bố Poisson để tính trong 1000 trang sách có 100 lỗi in sai. Tính xác suất để khi lấy ngẫu nhiên 1 trang có không quá 4 lỗi.

Bài giải Phân bố Poisson cho bởi biểu thức :

$$P(x,y) = \frac{a^m}{m!} e^{-a}$$

trong bài này $a = \frac{100}{1000} = 0,1$, $m = 4$

Chương trình gồm các bước sau :

- Đọc từ bàn phím số lỗi m = 4, a = 0,1
- Xây dựng hàm GIAI THUA (m)

- Dùng hàm mũ pow để tính a^m
- Tính phân bố Poisson theo công thức $\frac{a^m}{m!} e^{-a}$

```
/* CT83 Chuong trinh tinh phan bo Poisson */
#include<stdio.h>
#include<math.h>
float GIAITHUA(int);

void main()
{
int m;
float a,P;
printf("\n Cho m = ");scanf("%d",&m);
printf("\n Cho a = ");scanf("%f",&a);
P =pow(a,m)*exp(-a)/GIAITHUA(m);
printf("\n Phan bo Poisson P = %f",P);
getch();
return;
}

/* Ham giai thua */
float GIAITHUA(int n)
{
if(n==1||n==0)
    return(1);
else
    return(n*GIAITHUA(n-1));
}
```

Chương trình CT83 cho kết quả như sau :

Cho m = 4

Cho a = 0.1

Phan bo Poisson P = 0.000004

Bài số 84 Biến ngẫu nhiên x_i có xác suất p_i cho theo bảng phân bố x_i , p_i . Tìm kỳ vọng toán học, phương sai, độ lệch quân phương của các biến với số a cho trước bất kỳ.

Bài giải Kỳ vọng toán học của các biến ngẫu nhiên được xác định theo công thức :

$$M(x) = \sum_{i=1}^{n} x_i p_i$$

trong đó x_i là biến thứ i, p_i là xác suất xuất hiện biến x_i

Phương sai : $D(x) = M[x - M(x)]^2 = M[(x-a)^2] - (M-a)^2$

Độ lệch quân phương : $\sigma_x = \sqrt{D(x)}$

Chương trình gồm các bước sau :

- Đọc bảng x[i], p[i], số a, n, từ bàn phím.
- Tính kỳ vọng toán học M(x) theo thuật toán tính một tổng (xem CT23).
- Tính $D_1 = \sum_{i=1}^{n} p_i (x_i - a)^2$ và tính phương sai D(x) theo thuật toán tính một tổng.
- Cuối cùng tính độ lệch quân phương và ghi kết quả.

```
/* CT84 Chuong trinh tinh ky vong va phuong sai */
#include<stdio.h>
#include<math.h>
```

```
void main()
{
int i,n;
float d,d1,m,a,sigma;
float x[100],p[100];
printf("\n Cho so lan thu n = ");scanf("%d",&n);
printf("\n Cho so a = ");scanf("%f",&a);
printf(" Vao bang x[i] ");
for(i=1;i<=n;i++)
    {
    printf("\nx[%d] = ",i);scanf("%f",&x[i]);
    printf("\n p[%d] = ",i); scanf("%f",&p[i]);
    }
m = 0;
d1 = 0;
for (i=1;i<=n;i++)
    {
    m += x[i]*p[i];
    d1 += p[i]*(x[i]-a)*(x[i]-a);
    d = abs(d1-(m-a)*(m-a));
    sigma = sqrt(d);
    }
printf("\n Ky vong toan hoc m = %f",m);
printf("\n Phuong sai d = %f",d);
printf("\n Do lech quan phuong s =%f",sigma);
getch();
return;
```

Kết quả chương trình nếu cho bảng giá tr

x[1] = 1 *p[1] = 0.2*

x[2] = 2 *p[2] = 0.3*

x[3] = -1 *p[3] = 0.3*

x[4] = 2.5 *p [4] = 0.4*

x[5] = 1.5 *p[5] = 0.1*

a = 2

ta tìm được :

Ky vong toan hoc *m = 1.650000*

Phuong sai *d = 2.000000*

Do lech quan phuong *s = 1.414214*

Bài số 85 Xác định hệ số tương quan giữa hai biến ngẫu nhiên x và y. Kiểm tra xem mối liên hệ giữa x và y có đủ tin cậy không.

Bài giải Hai biến ngẫu nhiên x và y dạng vectơ n phần tử x(i) và y(i), $0 \le i \le n$ có tương quan với nhau. Để xác định hệ số tương quan giữa hai biến ta áp dụng công thức :

$$R_{xy} = \frac{C_{xy}}{\sigma_x \cdot \sigma_y}$$

trong đó : $C_{xy} = \frac{1}{n}\sum_{i=1}^{n} x_i y_i - \bar{x}\,\bar{y}$

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad , \qquad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

$$\sigma_x^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i^2 - \bar{x}^2) \ , \qquad \sigma_y^2 = \frac{1}{n}\sum_{i=1}^{n} (y_i^2 - \bar{y}^2)$$

và kiểm tra điều kiện $| R_{xy} | . \sqrt{n-1} \ge 3$ thì liên hệ giữa các biến x(i) và y(i) là tin cậy.

Chương trình gồm các công việc sau :

- Đọc các vectơ x(i) và y(i) từ bàn phím.
- Tính trung bình cộng $\bar{x}$ và $\bar{y}$ theo thuật toán tính một tổng.
- Tính các độ lệch quân phương σ_x và σ_y .
- Tính hệ số tương quan R_{xy} và kiểm tra điều kiện liên hệ tin cậy

$$| R_{xy} | . \sqrt{n-1} \geq 3$$

```
/* CT85 Chuong trinh tinh he so tuong quan */
#include<stdio.h>
#include<math.h>

void main()
{
int i,n;
float t,r,c,cxy,cxy1,sigmax,sigmay,rxy,
       sigmax1,sigmay1,x1,y1,xtb,ytb;
float x[100],y[100];
printf("\n He so tuong quan giua hai bang du lieu ");
printf("\n Vao du lieu ");
printf("\n Cho so bien n = "); scanf("%d",&n);
printf("\n Vao bang x[i],y[i]");
for(i=1;i<=n;i++)
    {
    printf("\n x[%d ] = ",i);
    scanf("%f",&x[i] ;
    printf("\n y[%d ] = ",i);
    scanf("%f",&y[i]);
    }
```

```
x1=0 ;y1=0;sigmax=0;sigmay=0;cxy=0;
for(i=1;i<=n;i++)
    {
    x1=x1+x[i];
    y1 = y1+y[i];
    }
xtb = x1/n;ytb = y1/n;
printf("xtb = %f ytb = %f",xtb,ytb);
for(i=1;i<=n;i++)
    {cxy = cxy+x[i]*y[i];}
cxy1 = cxy/n-xtb*ytb;
printf("\n cxy1 = %f ", cxy1);
for(i=1;i<=n;i++)
    { sigmax = sigmax+x[i]*x[i]-xtb*xtb;}
sigmax1 = sqrt(abs(sigmax/n));
printf("\n sigmax1 = %f",sigmax1);
for(i=1;i<=n;i++)
    { sigmay = sigmay+y[i]*y[i]-ytb*ytb;}
sigmay1 = sqrt(abs(sigmay/n));
printf("\n sigmay1 = %f",sigmay1);
rxy =cxy1/(sigmax1+sigmay1);
t = sqrt(n-1);
if(abs(rxy*t)>=3)
printf("\n Lien he tin cay rxy = %f",rxy);
else printf("\n Lien he khong tin cay rxy = %f",rxy);
getch();
return;
}
```

Kết quả chương trình nếu cho các bảng giá trị : n = 5

x[1] = 1 *y[1] = 2*

x[2] = -1 *y[2] = 3*

x[3] = 0 *y[3] = -1*

x[4] = 3 *y[4] = 1*

x[5] = 2 *y[5] = 2*

xtb = 1.000000 ytb = 1.400000

cxy1 = - 0.200000

sigmax1 = 1.414214

sigmay1 = 1.000000

Lien hẹ không tin cay rxy = - 0.082843

Chương 11

MỘT SỐ CHƯƠNG TRÌNH XỬ LÝ VĂN BẢN, QUẢN LÝ VÀ ỨNG DỤNG

Chương này trình bày một số chương trình xử lý văn bản, quản lý và một số bài toán ứng dụng của ngôn ngữ C. Phần giải thích chương trình đã được đưa vào văn bản chương trình nên lời giải thích dài trong chương này là không cần thiết.

Bài số 86 Lập trình in tờ lịch thế kỷ.

Bài giải Để có một tờ lịch cần các dữ liệu sau :

Thế kỷ, năm, tháng, số ngày trong tuần

Lịch chia làm hai phần. Phần một cần thử xem tháng có 30 hay 31 ngày. Tháng hai là tháng ngoại lệ. Trong phần hai chỉ có hai tháng có 30 ngày là tháng 9 và tháng 11, điều phân chia này làm đơn giản chương trình. Đồng thời cũng tính đến năm nhuận.

```
/* CT86 Chuong trinh lich the ky */
#include <stdio.h>
#include <conio.h>
#define N0 "CHU NHAT"
#define N1 "Thu Hai"
```

```
#define N2 "Thu Ba"
#define N3 "Thu Tu"
#define N4 "Thu Nam"
#define N5 "Thu Sau"
#define N6 "Thu Bay"
int ngaytuan;  /* 1..6 */
int ngay;      /* 1..31 */
int thang;     /* 1..12 */
int nam;       /* 1..3000 */
int theky;     /* 0..30 */
int NgayThu(int ,int ,int ,int );
void InDate(void);
int NamNhuan(int );

void main()
{
clrscr() ;
printf("\n\t\t\t\t\tLICH THE KY DEN NAM 3000");
printf("\nHay cho nam can lap LICH");
printf("\nNAM : ");
scanf("%d",&nam);
printf("\n\t\t\t\t\NAM : %6d\n\n",nam);
printf("\n Thang Gieng Thang Hai ");
printf(" Thang Ba Thang Tu ");
printf("' Thang Nam Thang Sau\n");
for (ngay=1;ngay<=31;ngay++)
    {
    for (thang=1;thang<=6;thang++)
```

```
        if ((thang%2)==0)
            if (thang==2)
                if (ngay>28)
                    if ((ngay==29) && (NamNhuan(nam)))
                        InDate;
                    else printf(" ");
                else InDate;
            else if (ngay==31) printf(" ");
                else InDate;
        else InDate;
    printf("\n");
    }
getch();
printf("\n");
printf("\n Thang Bay Thang Tam ");
printf(" Thang Chin Thang Muoi ");
printf(" Thang Mot Thang Chap\n");
for (ngay=1;ngay<=31;ngay++)
    {
    for (thang=7;thang<=12;thang++)
        if ((thang==9) || (thang==11)) && (ngay==31)
            printf(" ");
        else InDate;
            printf("\n");
    }
getch();
}
```

```
int NgayThu(int ng,int th,int n,int ngthu)
{
int nb,an;
if (th > 2) th = th-2;
else
    {
    th = th + 10;
    n = n - 1;
    }
an = n%100;
theky = (int)(n/100);
nb = (int)((13*th-1)/5)+(int)(an/4)+(int)(theky/4);
ngthu =((nb+an+ng-2*theky)%7);
return(ngthu);
}
void InDate(void)
{
NgayThu(ngay,thang,nam,ngaytuan);
switch (ngaytuan)
    {
    case 0 : printf(" %3d %9s",ngay,N0);
             break;
    case 1 : printf(" %3d %9s",ngay,N1);
             break;
    case 2 : printf(" %3d %9s",ngay,N2);
             break;
    case 3 : printf(" %3d %9s",ngay,N3);
             break;
```

```
    case 4 :  printf(" %3d %9s",ngay,N4);
              break;
    case 5 :  printf(" %3d %9s",ngay,N5);
              break;
    case 6 :  printf(" %3d %9s",ngay,N6);
              break;
    }
getch();
}
int NamNhuan(int n)
{
if ((n%4)==0) return(1);
    else return(0);
}
```

Bài số 87 Xây dựng hàm để kiểm tra dữ liệu vào có phải là số nguyên hay không.

Bài giải Ta xây dựng 3 hàm kiểm tra việc vào dữ liệu :

*void VaoSo1(int * p)* có đối số là kiểu nguyên, trả về một void.

int VaoSo2(void) có đối số kiểu void, trả về một kiểu nguyên.

*int VaoSo3(char * thongbao, int * p, int sai_max)* có điểm số lần vào sai cho phép.

```
/* CT87 Ham de quy - kiem tra so lieu vao la so nguyen */
#include <stdio.h>
#define max 10 /* so dai nhat co 10 chu so */
void VaoSo1(int *);
int VaoSo2(void);
int VaoSo3(char *,int *,int);
```

```
main()
{
int n;
const solanvaosai = 3;      /* Cho function VaoSo3
Neu vao sai > 3 lan thi stop chuong trinh - cho VaoSo3 *
printf("\nCho mot so (kieu 1) : ");
VaoSo1(&n);
printf("\nSo vao dung la : %d\n",n);
getch();
printf("\nCho mot so (kieu 2) : ");
n = VaoSo2();
printf("\nSo vao dung la : %d\n",n);
getch();

   /* Vao kieu thu 3 */
n = 0 ; /* Khoi tao lai n */
if (VaoSo3("Cho mot so (kieu 3) : ",&n,solanvaosai) != -1)
   printf("\nSo vao dung la : %d\n",n);
else printf("Vao sai qua tam ba ban - Stop");
getch();
return;
}
void VaoSo1(int *p)
{
int kiemtra;
char conso[max+1]; /* Dung cho lenh fgets ,
                      +1 de tinh den dau ket thuc dong \O */
fgets(conso,max,stdin);
```

```
kiemtra = sscanf(conso,"%d",p);
if (!kiemtra)
    {
    printf("\nVao sai ! Moi vao lai : ");
    VaoSo1(p);
    }
}
int VaoSo2(void)
{
int kiemtra,p;
char conso[max+1]; /* Dung cho lenh fgets
                     +1 de tinh den dau ket thuc dong \O */
fgets(conso,max,stdin);
kiemtra = sscanf(conso,"%d",&p);
if (!kiemtra)
    {
    printf("\nVao sai ! Moi vao lai : ");
    p = VaoSo2();
    }
return(p);
}
int VaoSo3(char *thongbao,int *p,int sai_max)
/* sai_max - so lan vao sai cho phep */
{
int kiemtra;
char conso[max+1]; /* Dung cho lenh fgets ,
                    +1 de tinh den dau ket thuc dong \O */
printf("%s",thongbao);
```

```
fgets(conso,max,stdin);
kiemtra = sscanf(conso,"%d",&p);
if (!kiemtra)
   if (--sai_max)
      return(VaoSo3("Vao sai ! Moi vao lai : ",p,sai_max));
   else return(-1);
else return(0);
}
```

Kết quả CT87 như sau :

Cho mot so (kieu 1) : 15

So vao dung la : 15

Cho mot so (kieu 2) : 25.75

So vao dung la 25

Cho mot so (kieu 3) : a5

Vao sai ! Moi vao lai :

Bài số 88 Lập trình một tờ hóa đơn

Bài giải Một tờ hóa đơn thường có các mục sau đây :

- Số thư tự
- Tên loại hàng
- Số lượng hàng
- Đơn giá
- Giá tiền loại hàng
- Tổng giá tiền của hóa đơn.

Chương trình CT88 gồm việc vào dữ liệu dạng mảng chứa tên loại hàng, đơn giá, số lượng và cần thực hiện tính tiền từng loại hàng và tổng tiền của hóa đơn theo thuật toán tính tổng một dãy số (CT23) và in kết quả.

```
/* CT88 Chuong trinh lap hoa don */
#include <stdio.h>
#include <conio.h>
#include <ctype.h>
#define MAX 50 /* so chung loai hang */

void main()
{
int i,n;
char OK,Hang[11];   /* Ten hang so ky tu dai nhat la 11 */
float Dongia[MAX],Soluong[MAX],Tien[MAX];
float Tongtien;
clrscr();
i = 1;
OK = 'C';                /* Dung de xac dinh so chung loai hang */
do
    {
    printf("\nTen loai Hang[%d] : ",i);scanf("%s",Hang[i]);
    printf("\n Don gia[%d] : ",i);scanf("%f",&Dongia[i]);
    printf("\n So luong[%d] : ",i);scanf("%f",&Soluong[i]);
    i++;
    printf("\nCo vao hang khac nua khong (c/k) : ");
    getchar();
```

```
    if (toupper(getchar())=='C')
        OK = 'C';
    else
        OK = 'K';
    }
while (OK=='C');
n = i;
printf("\n\n CHUONG TRINH LAP HOA DON ");
printf("\n\n\n ------- HOA DON -------- ");
printf("\n * No * Loai hang * So luong * Don gia USD * Tien
         USD *");
i = 1;
Tongtien = 0;
while ( i < n)
    {
    Tien[i] = Dongia[i]*Soluong[i];
    Tongtien = Tongtien + Tien[i];
    printf("\n %3d %8s %15.2f %15.2f %15.2f",i,Hang[i],Soluong[i],
           Dongia[i],Tien[i]);
    i++;
    }
printf("\n\n Tong so tien : %25.2f ",Tongtien);
getch();
return;
}
```

Theo CT88 ta có tờ hóa đơn như sau :

---HOA DON---

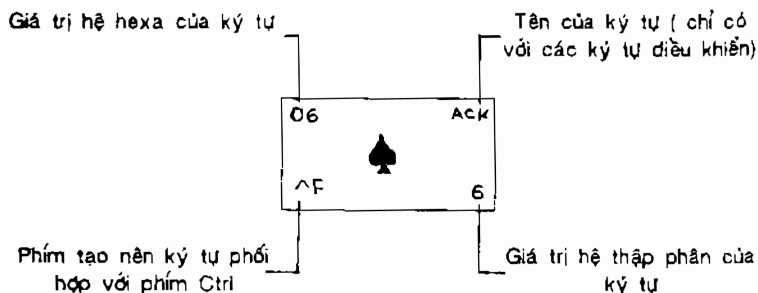
* No	* Loai hang	* So luong	* Don gia USD	* Tien USD
1	moto	2.00	1540.00	3080.00
2	radio	3.00	150.00	450.00
3	tivi	1.00	420.00	420.00

Tong so tien : 3950.00

Bài số 89 Lập trình in các kiểu kẻ khung ký tự đồ hoạ mã ASCII.

Bài giải

256 ký tự của mã ASCII được biểu diễn như sau :



Bằng cách bố trí các giá trị hệ hexa của các ký tự theo vị trí toạ độ màn hình (ở chế độ văn bản) ta có thể tạo nên những kiểu in theo ý muốn.

Các chương trình soạn thảo văn bản tiếng Việt như BKED, Vietstar cũng được xây dựng theo nguyên lý này.

```
/* CT89 Cac kieu ke khung dung ky tu do hoa ma ASCII */
#include <stdio.h>
#include <conio.h>
```

```
void main()
{
int i,k;
int *j
char a[255];
clrscr();
printf("\n\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB");
printf("\n\xDB\xDB \xDB\xDB");
printf("\n\xDB\xDB \xDB\xDB");
printf("\n\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB");
printf("\n");
printf("\n\xC9\xCD\xCD\xCB\xCD\xCD\xCB\xCD\xCD\xCD\xCD
       \xCB\xCD\xCD\xBB");
printf("\n\xBA \xBA \xBA \xBA \xBA");
printf("\n\xCC\xCD\xCD\xCE\xCD\xCD\xCE\xCD\xCD\xCD\xCD
       \xCE\xCD\xCD\xB9");
printf("\n\xBA \xBA \xBA \xBA \xBA");
printf("\n\xC8\xCD\xCD\xCA\xCD\xCD\xCA\xCD\xCD\xCD\xCD
       \xCA\xCD\xCD\xBC");
printf("\n");
/* Ke khung kep */

printf("\n\xC9");   /* Goc trai tren Alt+201 = "É" = so hex C9 */
for (i=1;i<79;i++)
   printf("\xCD"); /* In 78 net kep ngang N 206 "Í" = CD */
printf("\xBB");     /* Goc phai tren N 187 "»" = BB */
printf("\xBA");     /* Canh ben N 186 "º" = BA */
for (i=1;i<79;i++)
   printf(" ");
```

```
printf("\xBA");
printf("\xBA");
for (i=1;i<79;i++)
    printf(" ");
printf("\xBA");
printf("\xC8");     /* Goc trai duoi N 201 "È" = C8 */
for (i=1;i<79;i++)
    printf("\xCD");
printf("\xBC");     /* Goc phai duoi N 188 "¼" = BC */

/* Ke khung don */
printf("\n\xDA");
for (i=1;i<79;i++)
    printf("\xC4") ;
printf("\xBF");
printf("\xB3");
for (i=1;i<79;i++)
    printf(" ");
printf("\xB3");
printf("\xB3");
for (i=1;i<79;i++)
    printf(" ");
printf("\xB3");
printf("\xC0");
for (i=1;i<79;i++)
    printf("\xC4") ;
printf("\xD9");
printf("\n");
```

```
/* In bang ma ASCII */
printf("\nIn lan dau\n");
getch();
i=0;
{
printf("\n%d = \x00 %d = \x01 %d = \x02 %d = \x03 %d = \x04
%d = \x05 %d = \x06 %d = \x07 %d = \x08 %d = \x09 %d =\x0A",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);
printf("\n%d = \x0B %d = \x0C %d = \x0D %d = \x0E %d=\x0F
%d = \x10 %d = \x11 %d = \x12 %d = \x13 %d = \x14 %d =\x15",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);
printf("\n%d = \x16 %d = \x17 %d = \x18 %d = \x19 %d=\x1A
%d = \x1B %d = \x1C %d = \x17 %d = \x18 %d = \x19 %d=\x1A",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);
printf("\n%d = \x1B %d = \x1C %d = \x1D %d = \x1E %d=\x1F
%d = \x20 %d = \x21 %d = \x22 %d = \x23 %d = \x24 %d=\x25",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);
printf("\n%d = \x26 %d = \x27 %d = \x28 %d = \x29 %d =\x2A
%d = \x2B %d = \x2C %d = \x2D %d =\x2E %d =\x2F %d =\x30",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);
printf("\n%d = \x31 %d = \x31 %d = \x33 %d = \x34 %d = \x35
%d = \x36 %d = \x37 %d = \x38 %d = \x39 %d = \x3A %d =\x3B",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);
printf("\n%d = \x3C %d = \x3D %d = \x3E %d = \x3F %d =
\x40 %d = \x41%d =\x42 %d =\x43 %d =\x44 %d =\x45 %d=\x46",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);
getch();
printf("\n%d =\x47 %d = \x48 %d =\x49 %d =\x4A %d =\x4B
%d =\x4D %d =\x4E %d =\x4F %d =\x50 %d =\x51 %d=\x52",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);
```

```
printf("\n%d = \x53 %d = \x54 %d = \x55 %d = \x56 %d = \x57
%d = \x58 %d = \x59 %d = \x5A %d = \x5B %d =\x5C %d =\x5D"'
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

printf("\n%d = \x5E %d = \x5F %d = \x60 %d = \x61 %d =\x62
%d = \x63 %d = \x64 %d = \x65 %d = \x66 %d = \x67 %d =\x68".
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

printf("\n%d = \x69 %d = \x6A %d = \x6B %d =\x6D %d =\x6E
%d = \x6F %d = \x70 %d = \x71 %d = \x72 %d = \x73 %d =\x74",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

printf("\n%d = \x75 %d = \x76 %d = \x77 %d = \x78 %d =\x79
%d = \x7A %d = \x7B %d = \x7C %d =\x7D %d =\x7E %d =\x7F",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

printf("\n%d = \x80 %d = \x81 %d = \x82 %d = \x83 %d = \x84
%d = \x85 %d = \x86 %d = \x87 %d = \x88 %d = \x89 %d =\x8A",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

getch();

printf("\n%d = \x8B %d = \x8C %d =\x8D %d =\x8E %d =\x8F
%d = \x90 %d = \x91 %d = \x92 %d = \x93 %d = \x94 %d =\x95",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

printf("\n%d = \x96 %d = \x97 %d = \x98 %d =\x99 %d =\x9A
%d = \x9B %d = \x9C %d =\x9D %d =\x9E %d =\x9F %d =\xA0",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

printf("\n%d = \xA1 %d = \xA2 %d = \xA3 %d = \xA4 %d =
\xA5 %d = \xA6 %d = \xA7 %d = \xA8 %d = \xA9 %d = \xAA",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1.i+=1,i+=1,i+=1);

printf("\n%d = \xAB %d =\xAC %d =\xAD %d =\xAE %d =\xAF
%d =\xB0 %d =\xB1 %d =\xB2 %d =\xB3 %d =\xB4 %d=\xB5",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

printf("\n%d = \xB6 %d = \xB7 %d =\xB8 %d =\xB9 %d =\xBA
%d = \xBB %d =\xBC %d =\xBD %d =\xBE %d =\xBF %d =\xC0",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);
```

```
printf("\n%d = \xC1 %d = \xC2 %d = \xC3 %d =\xC4 %d =\xC5
%d = \xC6 %d = \xC7 %d = \xC8 %d = \xC9 %d = \xCA",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

getch();

printf("\n%d =\xCB %d =\xCC %d =\xCD %d =\xCE %d =\xCF
%d = \xD0 %d =\xD1 %d =\xD2 %d = \xD3 %d = \xD4 %d=\xD5",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

printf("\n%d = \xD6 %d = \xD7 %d =\xD8 %d =\xD9 %d =\xDA
%d =\xDB %d =\xDC %d =\xDD %d =\xDE %d =\xDF %d =\xE0",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

printf("\n%d = \xE1 %d = \xE2 %d = \xE3 %d =\xE4 %d =\xE5
%d = \xE6 %d = \xE7 %d = \xE8 %d = \xE9 %d = \xEA",
i+=1,i+=1,i+=1.i+=1,i+=1,i+=1,i+=1.i+=1,i+=1,i+=1);

printf("\n%d = \xEB %d =\xEC %d = \xED %d =\xEF %d =\xF0
%d = \xF1 %d = \xF2 %d = \xF3 %d = \xF4 %d =\xF5 %d =\xF6",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

printf("\n%d = \xF7 %d = \xF8 %d = \xF9 %d =\xFA %d =\xFB
%d = \xFC %d = \xFD %d = \xFE %d = \xFF",
i+=1,i+=1,i+=1.i+=1,i+=1,i+=1,i+=1,i+=1,i+=1) ;

}

getch();


printf("\nIn lan hai\n");

printf("\n\n\x00 \x01 \x02 \x03 \x04 \x05 \x06 \x07 \x08 \x09
\x0A");

printf("\n\n\x0B \x0C \x0D \x0E \x0F \x10 \x11 \x12 \x13 \x14
\x15");

printf("\n\n\x16 \x17 \x18 \x19 \x1A \x1B \x1C \x17 \x18 \x19
\x1A");

printf("\n\n\x1B \x1C \x1D \x1E \x1F \x20 \x21 \x22 \x23 \x24
\x25");
```

```
    printf("\n\n\x26 \x27 \x28 \x29 \x2A \x2B \x2C \x2D \x2E \x2F
\x30");
    printf("\n\n\x31 \x31 \x33 \x34 \x35 \x36 \x37 \x38 \x39 \x3A
\x3B");
    getch();
    printf("\n\n\x3C \x3D \x3E \x3F \x40 \x41 \x42 \x43 \x44 \x45
\x46");
    printf("\n\n\x47 \x48 \x49 \x4A \x4B \x4D \x4E \x4F \x50 \x51
\x52");
    printf("\n\n\x53 \x54 \x55 \x56 \x57 \x58 \x59 \x5A \x5B \x5C
\x5D");
    printf("\n\n\x5E \x5F \x60 \x61 \x62 \x63 \x64 \x65 \x66 \x67
\x68");
    printf("\n\n\x69 \x6A \x6B \x6D \x6E \x6F \x70 \x71 \x72 \x73
\x74");
    printf("\n\n\x75 \x76 \x77 \x78 \x79 \x7A \x7B \x7C \x7D \x7E
\x7F");
    printf("\n\n\x80 \x81 \x82 \x83 \x84 \x85 \x86 \x87 \x88 \x89
\x8A");
    getch();

    printf("\n\n\x8B \x8C \x8D \x8E \x8F \x90 \x91 \x92 \x93 \x94
\x95");
    printf("\n\n\x96 \x97 \x98 \x99 \x9A \x9B \x9C \x9D \x9E \x9F
\xA0");
    printf("\n\n\xA0 \xA1 \xA2 \xA3 \xA4 \xA5 \xA6 \xA7 \xA8 \xA9
\xAA");
    printf("\n\n\xAB \xAC \xAD \xAE \xAF \xB0 \xB1 \xB2 \xB3 \xB4
\xB5");
    printf("\n\n\xB6 \xB7 \xB8 \xB9 \xBA \xBB \xBC \xBD \xBE \xBF
\xC0");
```

```
    printf("\n\n\xC0 \xC1 \xC2 \xC3 \xC4 \xC5 \xC6 \xC7 \xC8 \xC9
\xCA");
    printf("\n\n\xCB \xCC \xCD \xCE \xCF \xD0 \xD1 \xD2 \xD3 \xD4
\xD5");
    getch();

    printf("\n\n\xD6 \xD7 \xD8 \xD9 \xDA \xDB \xDC \xDD \xDE
\xDF \xE0");
    printf("\n\n\xE0 \xE1 \xE2 \xE3 \xE4 \xE5 \xE6 \xE7 \xE8 \xE9
\xEA");
    printf("\n\n\xEB \xEC \xED \xEF \xF0 \xF1 \xF2 \xF3 \xF4 \xF5
\xF6");
    printf("\n\n\xF7 \xF8 \xF9 \xFA \xFB \xFC \xFD \xFE \xFF") ;
    getch();

    for (i=0;i<=256;i++)
        {
        printf("%d = %c ",i,a[i]);
        }

    /*printf("\nIn lan ba\n");
    j = 0x00;
    for (k=0x00;k<=0xFF;k++,j++)
        {
        printf("%d = %x",k,*j);
        print(" ");
        }*/
    getch();
    }
```

Bài số 90 Lập trình vào_ ra một dữ liệu kiểu cấu trúc.

Bài giải Xây dựng một cấu trúc có trường là một char c và x, y là hai số nguyên là tọa độ của điểm biểu diễn dạng :

```
struct diem
    {
    char c;
    int x, y;
    };
```

Xây dựng hai hàm void HienThi1 truy nhập kiểu thông thường và void HienThi2 truy nhập kiểu con trỏ và một cấu trúc để đảo ngược tọa độ x, y của điểm biểu diễn.

```
/* CT90 Kieu cau truc - Vao ra mot du lieu struct */
#include <stdio.h>
#include <conio.h>
struct diem
    { char c;
    int x,y;
    };
void HienThi1(struct diem p);      /* Bien thong thuong */
void HienThi2(struct diem *);      /* Bien kieu con tro */
struct diem Dao(struct diem *);  /* Ham dao toa do x,y */

main()
{
struct diem s1,s11,s22;
struct diem s2={'B',333,555};     /* Khoi tao bien cau truc s2 */
clrscr();
```

```
printf("\n%25cKieu struct cua ngon ngu C",'.');   /* Lui tu dau dong
                                                     25 cot */
s1.c = 'A';     /* Gan cac gia tri cho bien cau truc s1 */
s1.x = 111;
s1.y = 222;
HienThi1(s1);
getch();
HienThi2(&s2);
getch();
printf("\nDao nguoc toa do cac diem da cho");
s11 = Dao(&s1);     /* Dao nguoc toa do s1 */
HienThi2(&s11);     /* Vi ham Dao cho ket qua kieu con tro ,
                    nen khi hien thi phai dung kieu con tro */
printf("Diem goc s1 = %c %d %d\n",s1.c,s1.x,s1.y);
printf("Diem dao s11 = %c %d %d\n",s11.c,s11.x,s11.y);
getch();
s22 = Dao(&s2);
HienThi2(&s22);
printf("Diem goc s2 = %c %d %d\n",s2.c,s2.x,s2.y);
printf("Diem dao s22 = %c %d %d\n",s22.c,s22.x,s22.y);
getch();
return;
}

void HienThi1(struct diem p) /* Truy nhap kieu thong thuong */
{
 rintf("\nDiem %c co toa do la %d %d\n",p.c,p.x,p.y);
```

```
void HienThi2(struct diem *diachip) /* Truy nhap kieu con tro */
{
printf("\nDiem %c co toa do la %d %d\n", diachip->c, diachip->x,
diachip->y);
}
struct diem Dao(struct diem *diachiCA) /* Ham dao nguoc toa
do */
{
struct diem CA;              /* Diem co gia tri nghich dao */
CA.c = diachiCA->c;
CA.x = - diachiCA->x;
CA.y = - (*diachiCA).y;     /* Thay diachiCA.y = (*diachiCA).y */
return CA;      /* Gia tri dua ra cua ham : viet return CA hoac
return(CA) */
}
```

Ta được kết quả chương trình CT90 như sau:

Kieu struct cua ngon ngu C

Diem A co toa do la 111 222

Diem B co toa do la 333 555

Dao nguoc toa do cac diem da cho

Diem A co toa do la -111 -222

Diem goc s1 = A 111 222

Diem dao s11 = A -111 -222

Diem B co toa do la 333 555

Diem goc s2 = B 333 555

Diem dao s22 = B -333 -555

Bài số 91 Chương trình tháp Hà nội

Bài giải Chương trình tháp Hà nội là một ví dụ điển hình của thuật toán đệ quy của hàm, nghĩa là một hàm có thể gọi chính nó. Việc đệ quy cần sử dụng bộ nhớ kiểu xếp chồng LIFO (Last In, First Out stack) để chứa các kết quả trung gian do đó cần thận trọng khi kết thúc quá trình đệ quy.

Bài toán tháp Hà nội phát biểu như sau :

Cho ba chiếc cọc đánh số 1, 2, 3 và n chiếc đĩa có kích thước khác nhau Ban đầu các đĩa được xếp trên coc 1 với thứ tự đường kính giảm dần. Cần chuyển n đĩa từ cọc số 1 sang cọc số 3 theo nguyên tắc :

- Mỗi lần chỉ được chuyển một đĩa.

- Mỗi đĩa có thể được chuyển từ một cọc này sang cọc khác bất kỳ

- Không được đặt một đĩa trên đĩa khác có đường kính nhỏ hơn.

Với n = 1 thuật toán rất đơn giản, chỉ cần chuyển duy nhất một đĩa từ cọc 1 sang cọc số 3.

Khi n > 1 ta cần sử dụng cọc số 2 để chuẩn bị bước trung gian. Có thể phân tích bước trung gian này thành 2 thao tác đơn giản hơn :

- Chuyển n - 1 đĩa phía trên của cọc số 1 sang cọc số 2, trong giai đoạn này có thể sử dụng cọc số 3 như cọc trung gian.

- Chuyển n - 1 đĩa từ cọc 2 sang cọc số 3, khi đó có thể sử dụng cọc 1 như cọc trung gian.

Điều đó dẫn tới việc xây dựng một hàm đệ quy có các đối số sau đây:

- Số lượng đĩa cần chuyển

- Số cọc xuất phát

- Số cọc chuyển tới

- Số cọc trung gian

```
/* CT91 Chuong trinh thap Ha noi */
#include<stdio.h>
void main()
{
void hanoi (int,int,int,int);
int nd;            /* tong so dia */
printf("\n Co bao nhieu dia ? ");scanf("%d",&nd);
hanoi( nd,1,3,2);
}

/* ham de quy giai bai toan thap Ha noi */
void hanoi (int n,int xuat , int toi,int trung)
    /* n : so dia can chuyen */
    /* xuat : thap xuat phat */
    /* toi : thap can toi */
    /* trung : thap trung gian */
{
if (n > 0)
    {
    hanoi( n-1, xuat,trung,toi);
    printf(" Chuyen dich tu coc %d toi coc %d\n",xuat,toi);
    hanoi (n-1,trung,toi,xuat);
    }
getch();
return;
}
```

Kết quả CT91 như sau :

Co bao nhieu dia ? 3

chuyen dich	*tu coc*	*1*	*toi coc*	*3*
		1		*2*
		3		*2*
		1		*3*
		2		*1*
		2		*3*
		1		*3*

Chú ý khi n = 64 cần thời gian chuyển mất tới 58 tỉ năm !

Bài số 92 Mảng một chiều kiểu cấu trúc có các điểm là tọa độ của một đường cong.

Bài giải Ta xác định cấu trúc có tên là điểm tương tự CT90 với các tọa độ x, y của đường cong được nhập từ bàn phím và cấu trúc mảng các điểm và hiển thị tọa độ của các điểm.

```
/* CT92 Mang mot chieu kieu cau truc -
       so lieu cac diem cua mot duong cong */
#include <stdio.h>
#include <conio.h>
#define n 3            /* So diem cua mot duong cong */
#define maxline 128 /* Mot dong du lieu nhieu nhat 128 ky tu */
struct diem
    { char c;
    int x,y;
    };
struct diem curve[n];                    /* mang cac diem */
void HienThi1(struct diem [],int);       /* Kieu struct */
void HienThi(char [],int [],int [],int); /* Kieu don gian */
```

```
main()
{
int i;
char line[maxline];   /* mang cac ky tu du lieu cua 1 diem */
   /* cac bien cho kieu don gian */
char Ten[n];
int xx[n],yy[n];
/* Chuong trinh voi struct */
clrscr();
printf("\nCho cac diem cua duong cong-Bien kieu struct");
for (i=1;i<=n;i++)
   {
   printf("\nTen (1 chu cai) va toa do cua diem %d :",i);
   gets(line) ;
   sscanf(line,"%c %d %d",&curve[i].c,&curve[i].x,&curve[i].y);
   }
HienThi1(curve,n);
getch();
/* Chuong trinh kieu don gian */
clrscr();
printf("\nCho cac diem cua duong cong-Kieu don gian");
for (i=1;i<=n;i++)
   {
   printf("\nTen (1 chu cai) va toa do cua diem %d :",i);
   gets(line);
   sscanf(line,"%c %d %d",&Ten[i],&xx[i],&yy[i]);
   }
HienThi(Ten,xx,yy,n);
```

```
getch();
return;
}

void HienThi1(struct diem curve[],int a)
{
int i;
for (i=1 ;i<=a ;i++)
    printf("\nDiem %c toa do %d %d\n",curve[i].c,curve[i].x,
curve[i].y);
}

/* Ham cho kieu don gian */
void HienThi(char o[],int p[],int q[],int a)
{
int i;
for (i=1;i<=a;i++)
    printf("\nDiem %c co toa do la %d %d\n",o[i],p[i],q[i]);
}
```

Kết quả chương trình CT92 như sau :

Cho cac diem cua duong cong - Bien kieu struc

Ten (1 chu cai) va toa do cua diem 1 : a 12

Ten (1 chu cai) va toa do cua diem 2 : b 56

Ten (1 chu cai) va toa do cua diem 3 : c 25

Diem a co toa do 12 0

Diem b co toa do 56 0

Diem c co toa do 25 0

Bài số 93 Xây dựng một cấu trúc nhân sự để nhập vào danh sách:

- Họ tên
- Ngày vào cơ quan
- Ngày ra khỏi cơ quan

Bài giải Xây dựng hai cấu trúc có tên là nguời và date có trường là :

- Họ tên
- Ngày vào cơ quan
- Ngày ra khỏi cơ quan

và một hàm có tên là VaoDS để vào danh sách từng người

```
/* CT93 Bang kieu cau truc -danh sach tung nguoi */
#include <stdio.h>
#include <conio.h>
#define maxTen 40      /* Ten dai nhat la 40 ky tu */
struct date
   {
   int ngay;int thang;int nam;
   };
struct nguoi
   {
   char Ten[maxTen];
   struct date ngayvao;     /* ngay vao co quan */
   struct date ngayra;      /* ngay ra khoi co quan */
   };
void VaoDS(struct nguoi *ds);
```

```
main()
{
struct nguoi nhvat;
clrscr();
VaoDS(&nhvat);
printf("\nHo ten : %s \nNgay lam : %d%d%d \nNgay thoi viec
%d%d%d",
    nhvat.Ten,
    nhvat.ngayvao.ngay,nhvat.ngayvao.thang,nhvat.ngayvao.nam,
    nhvat.ngayra.ngay,nhvat.ngayra.thang,nhvat.ngayra.nam);
getch();
return;
}

void VaoDS(struct nguoi *ds)
{
char OK;            /* Bien dung cho viec hoi dap */
printf("\nCho ten: ");
gets(ds->Ten);      /* Chu y khong kiem tra do dai Ten */
printf("\nNgay vao co quan : ");
scanf("%d %d %d",&ds->ngayvao.ngay,
                 &ds->ngayvao.thang,
                 &ds->ngayvao.nam);
printf("\nNgay vao = ngay thoi viec ? y/n : ");
getchar();          /* De xuong dong */
OK = getchar();
if (OK == 'y') ds->ngayra = ds->ngayvao;
else
    {
```

```
    printf("\nNgay thoi viec : ") ;
    scanf("%d %d %d",&ds->ngayra.ngay,
                    &ds->ngayra.thang,
                    &ds->ngayra.nam) ;
    }
}
```

Kết quả của chương trình CT93 như sau :

Cho tên : Le thu Ha

Ngay vao co quan : 12 5 1990

Ngay vao = ngay thoi viec ? y/n : n

Ngay thoi viec : 14 7 1995

Ho ten : Le thu Ha

Ngay lam : 12 5 1990

Ngay thoi viec : 14 7 1995

Bài số 94 Lập danh sách nhân sự.

Bài giải Để lập cấu trúc danh sách nhân sự dùng hai cấu trúc người và date tương tự như CT93 ở đây có thêm một số người trong danh sách và do đó tạo nên một bảng danh sách đầy đủ.

```
/* CT94 Bang kieu cau truc - Danh sach nhan su */
#include <stdio.h>
#include <conio.h>
#define maxTen 40  /* Ten dai nhat la 40 ky tu */
#define max 50     /* So nguoi nhieu nhat trong danh sach */
struct date
    {
```

```
    int ngay;int thang;int nam;
    };
struct nguoi
    {
    char Ten[maxTen];
    struct date ngayvao;     /* ngay vao co quan */
    struct date ngayra;      /* ngay ra khoi co quan */
    };
void VaoDS(struct nguoi *ds);
main()
{
int i,n;
struct nguoi nhvat[max];
clrscr();
printf("\nCho so nguoi trong danh sach n = ");
scanf("%d",&n);
for (i=1;i<=n;i++)
    {
    printf("\nNguoi thu %d : ",i);
    VaoDS(&nhvat[i]);
    }
clrscr();
for (i=1;i<=n;i++)
    {
    printf("\nNguoi thu %d",i);
    printf("\nHo ten : %s \nNgay vao lam :%d%d%d\nNgay thoi viec
d%d%d",
    nhvat[i].Ten,
```

```
    nhvat[i].ngayvao.ngay,nhvat[i].ngayvao.thang,nhvat[i].ngayvao.nam,
    nhvat[i].ngayra.ngay,nhvat[i].ngayra.thang,nhvat[i].ngayra.nam);
    }
getch();
return;
}
void VaoDS(struct nguoi *ds)
{
char OK;              /* Bien dung cho viec hoi dap */
int i ;
printf("\nCho ten nguoi : ");
scanf("%s",ds->Ten);
/* gets(ds->Ten);    /* Chu y khong kiem tra do dai Ten */
       /* Lenh nay thuong hay bi nhay qua */
printf("\nNgay vao co quan : ");
scanf("%d %d %d",&ds->ngayvao.ngay,
                 &ds->ngayvao.thang,
                 &ds->ngayvao.nam) ;
printf("\nNgay vao = ngay bi duoi viec ? y/n : ") ;
getchar();       /* De xuong dong */
OK = getchar();
if (OK == 'y') ds->ngayra = ds->ngayvao;
else
    {
    printf("\nNgay bi thoi viec : ");
    scanf("%d %d %d",&ds->ngayra.ngay,
                     &ds->ngayra.thang,
                     &ds->ngayra.nam);
```

Kết quả chương trình CT94 như sau :

Cho so nguoi trong danh sach 2

Nguoi thu 1

Ho ten : Le thu Thuy

Ngay vao lam : 15 8 1990

Ngay thoi viec : 0

Nguoi thu 2

Ho ten : Ho van Tam

Ngay vao lam : 12 9 1975

Ngay thoi viec : 15 10 1994

Bài số 95 Lập bảng danh sách nhân sự kiểu cấu trúc dòng.

Bài giải Xây dựng hai cấu trúc để tổ chức danh sách nhân sự gồm:

Họ và tên

Ngày sinh

Quê quán

Mức lương

và lập hai hàm để vào danh sách và ra khỏi danh sách nhân sự.

```
/* CT95 Bang kieu cau truc dong - Danh sach nhan su */
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#define chuoi 50      /* Dai nhat la 50 ky tu */
```

```
struct date
    {
    int ngay;int thang;int nam;
    };
typedef struct phantu
    {
    char Ten[chuoi];
    struct date namsinh;
    char que[chuoi];
    int luong;
    struct phantu *tiep; /* kieu struct cap hai la con tro
        tro den phan tu tiep theo */
    } t_phantu; /* Kieu struct : phantu,bien struct : t_phantu */
int sott,songuoi;
void VaoDS(t_phantu **);      /* Bien con tro tro den mot con tro
khac */
void RaDS(t_phantu *);

main()
{
t_phantu *khoidau:      /* Con tro tro den dau danh sach */
clrscr();
VaoDS(&khoidau);
RaDS(khoidau);
getch();
return;
}
```

```
void VaoDS(t_phantu **qq)
{
char tenvao[chuoi],quevao[chuoi];
t_phantu *hientai;    /* De thay doi gia tri cua con tro */
clrscr();
*qq = NULL;           /* Khoi dau danh sach rong */
songuoi = 0;
while (1)    /* Vong lap vao danh sach khong han che so luong */
   {
   printf("Cho ten nguoi,khong cho nua thi go Enter\n");
   printf("Ten nguoi : ");
   gets(tenvao);
   if (strlen(tenvao))     /* Neu co vao ten nguoi */
      {
      songuoi += 1;
      hientai = (t_phantu *)malloc(sizeof(t_phantu));
      strcpy(hientai->Ten,tenvao);
      printf("Nam sinh (cac so cach nhau) : ");
      scanf("%d %d %d",&hientai->namsinh.ngay,
         &hientai->namsinh.thang,
         &hientai->namsinh.nam);
      printf("Que quan : ");
      gets(quevao);
      gets(quevao);
      strcpy(hientai->que,quevao);
      printf("Luong : ");
      scanf("%d",&hientai->luong);
```

```
        getchar(); /* De xuong dong sau khi vao xong mot nguoi */
        hientai->tiep = *qq;
        *qq = hientai;
        }
    else break; /* Ra khoi vong lap vao DS khi khong cho ten nua */
    } /* while */
}

void RaDS(t_phantu *khoidau)
{
clrscr();
printf("\n\t\t\t\t\t**** Danh sach ****") ;
printf("\n\n\ No\t Ho va ten \tNgay sinh\t\tQue quan\t\tLuong\n");
sott = 1;
while (khoidau)
    {
    printf(" %d%15s\t%d %d %d\t%15s\t%d\n",
        sott,khoidau->Ten,
        khoidau->namsinh,khoidau->que,khoidau->luong);
/* Muon hien thi dep nen chuyen doi so lieu thanh string roi moi
in */
    sott += 1;
    khoidau = khoidau->tiep;
    }
printf("\nTong so nguoi trong danh sach la %d",songuoi);
}
```

Kết quả chương trình CT95 như sau :

Cho ten nguoi, khong cho nua go Enter

**** Danh sach ****

No	Ho va ten	Ngay sinh	Que quan	Luong
1	Le thu Thuy	24 12 1979	Bat trang Gia Lam	100000
2	Nguyen van Long	12 6 1990	Pho Hue Ha noi	200000
3	Nguyen dieu Hoa	15 10 1985	Ba vi Ha noi	250000

Tong so nguoi trong danh sach la 3

Bài số 96 Tìm biểu tượng chòm sao theo ngày và tháng sinh.

Bài giải Theo lịch một số nước phương Tây có 12 chòm sao tương ứng với ngày sinh và tháng sinh sau đây :

Capricorne	23 tháng chạp đến 19 tháng giêng
Verseau	20 tháng giêng đến 19 tháng hai
Poisson	20 tháng hai đến 20 tháng ba
Bélier	21 tháng ba đến 19 tháng tư
Taurau	20 tháng tư đến 20 tháng năm
Gémeau	21 tháng năm đến 20 tháng sáu
Cancer	21 tháng sáu đến 21 tháng bảy
Lion	22 tháng bảy đến 22 tháng tám
Vierge	23 tháng tám đến 22 tháng chín
Balance	23 tháng chín đến 22 tháng mười
Scorpion	23 tháng mười đến 21 tháng mười một
Sagittaire	22 tháng mười một đến 22 tháng chạp

Chương trình cần dự tính xác định mối tương quan giữa tên một biểu tượng chòm sao và hai giá trị thời gian ngày sinh và tháng sinh tương ứng. Có thể tiến hành công việc kể trên theo các cách sau :

- Dùng nhiều mảng (ngày, tháng, chòm sao) liên hệ với nhau qua một chỉ số chung.

- Bằng một mảng cấu trúc mỗi phần tử gồm một số ứng với ngày sinh, một tên ứng với tháng sinh và một tên đặc trưng cho biểu tượng chòm sao. Phương án này cho ta mối liện hệ rõ ràng giữa các dữ liệu ở thời điểm khởi đầu chương trình. Việc tìm chòm sao tương ứng với ngày sinh và tháng sinh tiến hành như sau :

+ Đầu tiên tìm phần tử thuộc mảng cấu trúc có tên tháng ứng với dữ liệu đã cho, nếu không tương ứng sẽ hiển thị bằng dòng " sai tên tháng "

+ Tiếp theo sẽ tìm số ngày nhỏ hơn hoặc bằng ngày của phần tử tìm kiếm X.

Nếu ngày đó trước ngày cuối của biểu tượng ứng với phần tử X điều đó cho ta kết quả mong muốn.

+ Ngược lại ta có thể kết luận được rằng ngày đó sau ngày bắt đầu của biểu tượng ứng với phần tử X, ta chỉ cần xem xét phần tử tiếp theo để tìm phần tử mong muốn. Tuy nhiên nếu X là phần tử cuối của mảng thì tiếp theo nó phải là phần tử đầu của mảng.

- Chương trình CT96 sử dụng cấu trúc s_date trong đó có một số ứng với ngày sinh tên tháng sinh và biểu tượng chòm sao tương ứng. Ta trù tính 10 ký tự cho tên tháng cho phép một string nhỏ hơn hoặc bằng 9 ký tự (Lưu ý ký tự \O là ký tự kết thúc) và 11 ký tự dành cho tên biểu tượng chòm sao.

Mảng date là 12 phần tử, mỗi phần tử kiểu s-date đã được khởi đầu điều đó cho phép dễ dàng đặt song song mỗi biểu tượng và thời gian cuối của biểu tượng.

Hàm stricmp cho phép so sánh xâu ký tự không phân biệt chữ in và chữ thường

Việc tìm tháng sinh được thực hiện bằng một lệnh :

```
while ( stricmp ( date[i].thang, thang-s ) &&i ++< 11 ){}
```

vừa ngắn gọn, vừa cho phép biết ngay việc tìm kiếm có kết quả hay không. Cuối lệnh này i bằng 12 khi i < 12, i chỉ phần tử tìm thấy.

Việc sử dụng toán tử số học % cho phép giải quyết vấn đề biểu tượng tiếp theo biểu tượng cuối cùng của mảng.

Ở đây việc tìm kiếm tên tháng được thực hiện bằng thuật toán tìm kiếm tuần tự theo mảng.

```
/* CT96 Chuong trinh tim sao theo ngay sinh */
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
struct s_date
    {
    int ngay;
    char thang[10];
    char sao[12];
    }; /* Ten 12 chom sao */
struct s_date date[12] = {
                    23, "chap ", "Sagittaire",
                    20, "gieng", "Capricorne",
                    20, "hai" , "Verseau" ,
                    21, "ba" , "Poisson" ,
                    20, "tu" , "Belier" ,
                    21, "nam" , "Taureau" ,
                    21, "sau" , "Gemeau" ,
```

```
                22, "bay" , "Cancer" ,
                23, "tam" , "Lion" ,
                23, "chin" , "Vierge" ,
                23, "muoi" , "Balance" ,
                22, "mot" , "Scorpion"
                };
int ngay_s;
char thang_s[10];
int i;
printf("\n Ban cho biet ngay va thang sinh ? :");
scanf(" %d %s",&ngay_s,thang_s);
printf("\n Ngay sinh : %d",ngay_s);
printf("\n Thang sinh : %s",thang_s);

/* Tim kiem va hien thi chom sao tuong ung */
i = 0;
while (stricmp(date[i].thang,thang_s) && i++ <11) {}
if (i < 12)
   {
   printf("\n Ban duoc sinh duoi chom sao : ");
   if ( ngay_s >= date[i].ngay) i = (i+1)%12;
   printf("%s", date[i].sao );
   }
else printf(" **** Sai ten thang **** ");
getch();
return;
}
```

Trong chương trình CT96 nếu cho các dữ liệu ngày và tháng sinh ta sẽ được kết quả như sau :

Ban cho biet ngay va thang sinh ? : 14 muoi

Ngay sinh : 14

Thang sinh : muoi

Ban duoc sinh duoi chom sao : Balance

Bài số 97 Lập chương trình file truy nhập tuần tự.

Bài giải Chương trình cho phép hiển thị lần lượt các danh sách nhân sự như các chương trình CT94, CT95 kèm theo một số thứ tự trong danh sách.

```
/* CT97 Chuong trinh file truy nhap tuan tu */
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define maxten 30        /* Ten nhat 20 ky tu */
#define maxcon 15        /* So con nhieu nhat */
#define maxtenfile 40    /* Ten file va duong dan max 40 ky tu */
struct nguoi
    {
    char ten[maxten+1];/* +1 de tinh den ca ky tu ket thuc \O */
    int tuoi;
    int socon;
    int tuoicon[maxcon];
    } hoso;
void Hienthi(struct nguoi *,int);
```

```
main()
{
char tenfile[maxtenfile+1];    /* +1 de tinh den ca ky tu
            ket thuc \O */
FILE *f1;
int i,n;
/* Tao file */
printf("\nCho ten file : ");
gets(tenfile);
/* Hoac dung lenh fgets(tenfile,maxtenfile,stdin) de phong viec
      cho duong dan dai qua so ky tu maxtenfile */
if ((f1=fopen(tenfile,"w"))==NULL)
   {
   printf("\nLoi mo file - Chuong trinh ket thuc\n");
   exit(-1);
   }

/* Vao so lieu */
printf("Vao so lieu , muon thoi thi den muc Ten an Enter\n");
n = 0;   /* Dem so phan tu trong file */
do
   {
   printf("Ten : ");
   gets(hoso.ten);
   if (strlen(hoso.ten)==0) break;   /* Ket thuc neu ten la rong */
   n++;
   printf("Tuoi : ");
   scanf("%d",&hoso.tuoi);
   printf("So con : ");
```

```
    scanf("%d",&hoso.socon);
    for (i=0;i<hoso.socon && i<maxcon;i++)
        {
        printf("Tuoi con thu %d : ",i+1);
        scanf("%d",&hoso.tuoicon[i]);
        }
    getchar();   /* Xuong dong trong file sau mot nguoi */
    printf("\n");
    fwrite(&hoso,sizeof(hoso),1,f1);    /* Ghi vao file */
    }
while (1);
fclose(f1);
printf("\nKet thuc viec tao file va ghi du lieu vao.");
printf("\nTrong file nay co %d phan tu (nguoi)",n);
getch();
/* Hien thi noi dung file */
do
    {
    printf("\nCho ten file can doc : ") ;
    gets(tenfile) ;
    /* Hoac dung lenh fgets(tenfile,maxtenfile,stdin) de phong viec
            cho duong dan dai qua so ky tu maxtenfile */
    if ((f1=fopen(tenfile,"rb"))==0)
        printf("\nKhong tim thay file - Cho lai ten\n");
    }
while ( !f1);
n = 1;
while (fread(&hoso,sizeof(hoso),1,f1))
```

```
    {
    Hienthi(&hoso,n++);
    getchar();
    }
fclose(f1);
printf("\nDoc xong danh sach trong file .");
/* Them du lieu vao cuoi file */
do
    {
    printf("\nCho ten file can them du lieu : ");
    gets(tenfile);
    if ((f1=fopen(tenfile,"ab"))==0)  /* Mo de them vao cuoi file */
        printf("\nKhong tim thay file - Cho lai ten\n");
    }
while ( !f1);
do
    {
    printf("Ten : ") ;
    gets(hoso.ten) ;
    if (strlen(hoso.ten)==0) break ; /* Ket thuc neu ten la rong */
    n++ ;
    printf("Tuoi : ") ;
    scanf("%d",&hoso.tuoi) ;
    printf("So con : ") ;
    scanf("%d",&hoso.socon) ;
    for (i=0 ;i<hoso.socon && i<maxcon ;i++)
        {
        printf("Tuoi con thu %d : ",i+1) ;
```

```
        scanf("%d",&hoso.tuoicon[i]);
        }
    getchar();    /* Xuong dong trong file sau mot nguoi */
    printf("\n");
    fwrite(&hoso,sizeof(hoso),1,f1);    /* Ghi vao file */
    }
while (1);
fclose(f1);
printf("\nKet thuc viec ghi them du lieu vao cuoi file .");
getch();
return;
}

void Hienthi(struct nguoi *hoso,int so)
{
int i ;
printf("\nSo ho so : %d\n",so);
printf("Ten : %s\n",hoso->ten);
printf("Tuoi : %d\n",hoso->tuoi);
printf("So con : %d\n",hoso->socon);
for (i=0;i<hoso->socon && i<maxcon;i++)
    printf("Tuoi con thu %d : %d\n",i+1,hoso->tuoicon[i]);
}
```

Kết quả chương trình CT97 như sau :

Cho ten file : Danh sach

Vao so lieu muon thoi den muc Ten an Enter

Ten : Nguyen Kim Dung

Tuoi : 45

So con : 2

Tuoi con thu nhat : 21

Tuoi con thu hai : 15

Ket thuc viec tao file va ghi du lieu vao

Trong file nguoi co 1 phan tu (nguoi)

Cho ten file can doc : Danh sach

Doc xong danh sach trong file

Cho ten file can them du lieu

Ket thuc ghi len file

Bài số 98 Lập trình tạo file truy nhập tuần tự cho phép tạo, xem, sửa, thêm vào cuối danh sách và cho phép chọn tên danh sách.

Bài giải CT98 là chương trình đầy đủ tạo nên một danh sách hồ sơ cho phép.

- Tạo danh sách
- Hiển thị danh sách
- Sửa danh sách
- Thêm vào danh sách
- Đặt tên danh sách

Các lời giải thích đã nằm trong văn bản chương trình. Ta được kết quả của chương trình CT98 tương tự CT97 nhưng đầy đủ hơn.

```
/* CT98 Chuong trinh file truy nhap tuan tu - :
   tao,xem,sua,them vao cuoi file,cho phep chon ten file */
#include <stdio.h>
#include <conio.h>
```

```
#include <string.h>
#include <stdlib.h>
#define maxten 30       /* Ten dai nhat 30 ky tu */
#define maxcon 15       /* So con nhieu nhat */
#define maxtenfile 40   /* Ten file va duong dan max 40 ky tu */
struct nguoi
   {
   char ten[maxten+1]; /* +1 tinh ca ky tu ket thuc \O */
   int tuoi;
   int socon;
   int tuoicon[maxcon];
   } hoso;
char tenfile[maxtenfile+1];    /* +1 tinh ca ky tu ket thuc \O */
FILE *f1;
int n,     /* so ho so */
ngoai;    /* so > so ho so */
long sohoso,vitri;      /* vi tri hien tai trong file */
char dong[maxtenfile+1];
void TaoFile(void);
void Hienthi(struct nguoi *,int);
void Sua(struct nguoi *);
void Them(void);

main()
{
int i;
/* Mo file */
printf("\n*************** MO FILE ********************\n");
```

```
do
    {
    printf("\nCho ten file can mo : ");
    gets(tenfile);
    if ((f1=fopen(tenfile,"wt"))==NULL)         /* wt - mo de ghi
                                                    file Text */
        printf("\nLoi mo file - Cho lai ten file\n");
    }
while ( !f1);
TaoFile();
/* Hien thi noi dung file */
do
    {
    printf("\n************* XEM NOI DUNG FILE **********\n");
    printf("\nTen file dang soan : %s\n",tenfile);
    printf("\nCho ten file (neu la file dang soan thi an Enter) : ");
    gets(dong);     /* Cho ten moi */
    if (strlen(dong)) strcpy(tenfile,dong);
    if ((f1=fopen(tenfile,"rt"))==0)
        printf("\nKhong tim thay file - Cho lai ten\n");
    }
while ( !f1);
n = 1;
while (fread(&hoso,sizeof(hoso),1,f1))
    {
    Hienthi(&hoso,n++);
    getchar();
    }
```

```
fclose(f1);
printf("\nDoc xong danh sach trong file.\n\n");
/* Sua du lieu trong file */
do
   {
   printf("\n********* SUA CHUA NOI DUNG FILE *********\n");
   printf("\nTen file dang soan : %s\n",tenfile);
   printf("\nCho ten file (neu la file dang soan thi an Enter): ");
   gets(dong);      /* Cho ten moi */
   if (strlen(dong)) strcpy(tenfile,dong);
   if ((f1=fopen(tenfile,"r+t"))==0)
      printf("Khong tim thay file tren dia\n");
   }
while( !f1);
fseek(f1,0,2);
sohoso = ftell(f1) / sizeof(hoso);
/* Tim va sua ho so */
do
   {
   do
      {
      printf("\nCho so ho so can sua (go 0 de stop) : ");
      scanf("%d",&n);
      getchar();           /* De nhay qua ki tu cuoi \n */
      ngoai = n<0 || n>sohoso;
      }
   while (ngoai);
   if (n==0) break;     /* Khong sua ra khoi vong lap */
```

```
    vitri = (n-1)*sizeof(hoso);  /* Tinh vi tri cua ho so can sua */
    fseek(f1,vitri,0);   /* Dinh vi con tro den ho so */
    fread(&hoso,sizeof(hoso),1,f1); /* Doc mot ho so vao bo nho */
    Hienthi(&hoso,n); /* Hien thi noi dung ho so o bo nho */
    Sua(&hoso);    /* Sua ho so o bo nho */
    fseek(f1,vitri,0);
    fwrite(&hoso,sizeof(hoso),1,f1); /* Ghi ho so da sua vao file */
    }
while(1);
fclose(f1);
printf("\nDa sua xong\n\n");
/* Xem lai noi dung file da sua */
printf("\n********** XEM LAI NOI DUNG FILE ***********\n");
if ((f1=fopen(tenfile,"rt"))!=0)
    n = 1;
while (fread(&hoso,sizeof(hoso),1,f1))
    {
    Hienthi(&hoso,n++);
    getchar();
    }
fclose(f1);
printf("\nDoc xong danh sach trong file .\n\n");
getch();
/* Them du lieu vao cuoi file */
do
    {
    printf("\n******* THEM DU LIEU VAO CUOI FILE ******\n");
    printf("\nTen file dang soan : %s\n",tenfile);
```

```
    printf("\nCho ten file (neu la file dang soan thi an Enter) : ");
    gets(dong);     /* Cho ten moi */
    if (strlen(dong)) strcpy(tenfile,dong);
    if ((f1=fopen(tenfile,"at"))==0)  /* Mô file de them vao cuoi */
        printf("Khong tim thay file tren dia\n");
    }
while( !f1);
Them();
/* Hien thi noi dung file */
do
    {
    printf("\n*********** XEM NOI DUNG FILE ***********\n");
    printf("\nTen file dang soan : %s\n",tenfile);
    printf("\nCho ten file (neu la file dang soan thi an Enter) : ");
    gets(dong); /* Cho ten moi */
    if (strlen(dong)) strcpy(tenfile,dong);
    if ((f1=fopen(tenfile,"rt"))==0)
        printf("\nKhong tim thay file - Cho lai ten\n");
    }
while ( !f1);
n = 1;
while (fread(&hoso,sizeof(hoso),1,f1))
    {
    Hienthi(&hoso,n++);
    getchar();
    }
fclose(f1);
printf("\nDoc xong danh sach trong file.Enter to stop !");
```

```
getch();
return;
}

void TaoFile(void)
{
int i;
printf("Vao so lieu , muon thoi thi den muc Ten an Enter\n");
n = 0;   /* Dem so phan tu trong file */
do
    {
    printf("Ten      : ") ;
    gets(hoso.ten) ;
    if (strlen(hoso.ten)==0) break;   /* Ket thuc neu ten la rong */
    n++;
    printf("Tuoi     : ");
    scanf("%d",&hoso.tuoi);
    printf("So con  : ");
    scanf("%d",&hoso.socon);
    for (i=0;i<hoso.socon && i<maxcon;i++)
        {
        printf("Tuoi con thu %d : ",i+1);
        scanf("%d",&hoso.tuoicon[i]);
        }
    getchar();   /* Xuong dong trong file sau mot nguoi */
    printf("\n");
    fwrite(&hoso,sizeof(hoso),1,f1);   /* Ghi vao file */
    }
```

```
while (1);
fclose(f1);
printf("\nKet thuc viec tao file va ghi du lieu vao .");
printf("\nTrong file nay co %d phan tu (nguoi)",n);
getch();
}

void Hienthi(struct nguoi *hoso,int so)
{
int i;
printf("\nSo ho so : %d\n",so);
printf("Ten      : %s\n",hoso->ten);
printf("Tuoi     : %d\n",hoso->tuoi);
printf("So con  : %d\n",hoso->socon);
for (i=0;i<hoso->socon && i<maxcon;i++)
   printf("Tuoi con thu %d : %d\n",i+1,hoso->tuoicon[i]);
}

void Sua(struct nguoi *hoso)
{
int i;
printf("Sua chua so lieu , go Enter neu khong sua\n");
printf("Ten      : ");
gets(dong);      /* Cho ten moi */
if (strlen(dong)) strcpy(hoso->ten,dong);
printf("Tuoi     : ");
gets(dong);      /* Cho tuoi moi */
if (strlen(dong)) hoso->tuoi = atoi(dong);
printf("So con  : ");
```

```
gets(dong);      /* Cho so moi */
if (strlen(dong)) hoso->socon = atoi(dong);
for (i=0;i<hoso->socon && i<maxcon;i++)
    {
    printf("Tuoi con thu %d : ",i+1);
    gets(dong);      /* Cho so moi */
    if (strlen(dong)) hoso->tuoicon[i] = atoi(dong);
    }
}

void Them(void)
{
int i;
printf("Vao so lieu , muon thoi thi den muc Ten an Enter\n");
do
    {
    printf("Ten      : "); gets(hoso.ten);
    if (strlen(hoso.ten)==0) break;   /* Ket thuc neu ten la rong */
    n++;
    printf("Tuoi     : ");
    scanf("%d",&hoso.tuoi);
    printf("So con  : ");
    scanf("%d",&hoso.socon);
    for (i=0;i<hoso.socon && i<maxcon;i++)
        {
        printf("Tuoi con thu %d : ",i+1);
        scanf("%d",&hoso.tuoicon[i]);
        }
```

```
    getchar();    /* Xuong dong trong file sau mot nguoi */
    printf("\n");
    fwrite(&hoso,sizeof(hoso),1,f1);    /* Ghi vao file */
    }
  while (1);
  fclose(f1);
  printf("\nKet thuc viec them du lieu vao cuoi file.");
  getch();
  }
```

Bài số 99 Lập trình mã hóa morse và giải mã morse một bức điện đánh từ bàn phím.

Bài giải Ta biết mã morse gồm 26 ký tự chữ cái từ A đến Z (giả thiết đánh bằng chữ hoa), 10 con số từ 0 đến 9 và dấu chấm như sau:

BẢNG MÃ MORSE

A •–	B –•••	C –•–•	D –••	E •
F ••–•	G ––•	H ••••	I ••	J •–––
K –•–	L •–••	M ––	N –•	O –––
P •––•	Q ––•–	R •–•	S •••	T –
U ••–	V •••–	W •––	X –••–	Y –•––
Z ––••	. •–•–•–			
O –––––	1 •––––	2 ••–––	3 •••––	4 ••••–
5 •••••	6 –••••	7 ––•••	8 –––••	9 ––––•

Chương trình cần xác định sự tương ứng giữa các ký tự chữ và con số quen biết với các mã morse. Để làm việc này ta sử dụng một mảng cấu trúc gọi là code. Mỗi phần tử của mảng này gồm có :

- Một ký tự (từ A đến Z, từ 0 đến 9)

- Mã morse tương ứng với ký tự được biểu thị dưới dạng string các dấu • –

Ta xác định một struct có tên là code gồm một chữ và một pointer trỏ về một string chứa mã morse tương ứng.

Mảng chỉ chứa 37 phần tử (giá trị NL) kích thước nói chung gồm 5 byte 1 byte cho ký tự, 4 byte dành cho pointer

Quá trình dịch một bức điện bằng chữ sang mã morse tiến hành :

- Đọc bức điện từ bàn phím mỗi dòng cực đại 128 ký tự (chữ hoa).

- Việc tìm mã morse tương ứng được thực hiện bằng một lệnh :

```
while ( dong [i]! = mang [j] . chu && j ++ < NL - 1 );
```

Giải mã morse là quá trình ngược lại nghĩa là cần tìm một ký tự tương ứng với một mã morse. Để làm việc này chúng ta có thể sử dụng cùng một cấu trúc như đã làm với quá trình mã hóa, vì thế thuật toán tìm kiếm cũng tương tự. Quá trình giải mã morse cụ thể như sau :

- Đọc bức điện morse từ bàn phím bằng gets trong mảng cực đại 128 ký tự có tên là dong. Quá trình xử lý sau đây sẽ lặp lại và dựa trên việc sử dụng con trỏ char * morse :

- Tăng giá trị con trỏ khi có một khoảng cách.

- Nhờ sscanf lấy một string có độ dài lớn nhất là 7 (không có mã morse nào vượt quá kích thước này).

- Tăng giá trị con trỏ của string đã được đọc.

Việc tìm mã morse trong mảng được thực hiện chỉ bằng một lệnh :

```
while ( stricmp ( code, mang [ij . morse ) && j ++ < NL - 1 );
```

```
/* CT99 Chuong trinh ma hoa va giai ma morse */
#include<stdio.h>
#include<string.h>
#define NL 37    /* so ky tu duoc ma hoa */
#define LG 127   /* chieu dai dong morse */

void main()
{
struct code
    { char chu;
    char * morse;
    };
struct code mang[NL] =
    { 'A' , ".-" , 'B' , "-...", 'C' , "-.-." ,
    'D' , "-.." , 'E' , "." , 'F' , "..-." ,
    'G' , "--." , 'H' , "....", 'I' , ".." ,
    'J' , ".---" , 'K' , "-.-" , 'L' , ".-.." ,
    'M' , "--" , 'N' , "-." , 'O' , "---" ,
    'P' , ".--." , 'Q' , "--.-", 'R' , ".-." ,
    'S' , "..." , 'T' , "-" , 'U' , "..-" ,
    'V' , "...-" , 'W' , ".--" , 'X' , "-..-" ,
    'Y' , "-.--" , 'Z' , "--..", '.' , ".-.-.-" ,
    '0' , "-----" , '1' , ".----", '2', "..---" ,
    '3' , "...--" , '4' , "....-", '5', "....." ,
    '6' , "-...." , '7' , "--...", '8', "---.." ,
    '9' , "----." ,
    };
 char dong[LG+1];    /* de dóc 1 dong tu ban phim */
```

```
char code[7]; /* ma can dich */
int i,j
/* Doc buc dien */
printf("\n Cho noi dung cua buc dien 1 dong max 128 \n");
gets (dong);
printf("\n Day la ban dich ma morse cua buc dien\n ");
/* Dich buc dien tung chu */
for ( i = 0; i < strlen(dong); i++)
    {
    j = 0;
    while (dong[i] != mang[j].chu && j++ <NL-1);
    if (j <NL) printf( " %7s", mang[j].morse);
    else printf( "??????");
    if(! ((i+1)%10)) printf("\n");    /* 10 ma morse 1 dong */
    }
/* Doc buc dien morse can dich */
printf("\n Cho buc dien morse ( 1 dong max) :\n ");
gets(dong);
printf("\n Day la noi dung buc dien \n");
i = 0;
while (i < strlen(dong))
    {
    while(dong[i] == ' ') i++; /* nhay qua dau cach */
    if (i < strlen(dong))
        {
        sscanf (&dong[i], "%6s",code);      /* doc ma (max 6 ky tu */
        i += strlen(code);    /* tang con tro trong dong */
        j = 0;                /* tim ma trong mang */
```

```
        while(stricmp(code,mang[j].morse) && j++ <NL-1) ;
        if (j <NL) printf("%2c",mang[j].chu) ; /* ma tim thay */
        else printf(" ? ") ; /* ma khong thay */
        }
    }
}
```

Kết quả của chương trình CT99 như sau :

Cho noi dung cua buc dien 1 dong max 128

VO DE VE NGAY

Day la ban dich ma morse cua buc dien

..._ _ _ _ ?????? _.. .?????? .._ .?????? _.

_ _. ._ _._ _

Cho mot buc dien morse (1 dong max) :

..._ _ _ _ _.. . .._ . _. _ _. ._ _._ _

Day la noi dung cua buc dien

VO DE VE NGAY

Bài số 100 Lập trình cho một bản nhạc trên máy vi tính. Ghi bài Tiến quân ca của nhạc sỹ Văn Cao.

Bài giải Một nốt nhạc được đặc trưng bởi cao độ (tần số) và trường độ.

Có thể nhận được hiệu ứng âm thanh bằng việc lập trình bộ thời gian có thể lập trình PIT (Programmable Interval Timer) và bộ ghép nối ngoại vi có thể lập trình PI (Progammable Peripheral Interface).

Bằng việc lập trình PIT ta tác động lên tần số và khoảng kéo dài của âm thanh. Các hàm sau đây quyết định đến đặc tính âm thanh :

void sound (tanso) - để loa phát một âm với tần số cho trước.

void delay (thoigian) - để loa kéo dài âm phát ra trong khoảng thời gian cho trước.

void nosound (void) - để ngừng loa.

Trước hết ta xác định tần số của các nốt nhạc ứng với các octave, ở đây ta sử dụng 4 octave. Trường độ của nốt nhạc được xác định theo độ dài của nốt đen d và xác định độ dài của nốt trắng, nốt tròn, nốt móc đơn, v.v..., theo nốt đen. Cuối cùng gọi hàm bằng not (cao độ, trường độ) ; theo trật tự của bản nhạc. Dấu lặng nhận được bằng cách phát ra một siêu âm có tần số 30.000 Hz.

Chương trình CT100 ghi bản nhạc Tiến quân ca của nhạc sỹ Văn Cao, đồng thời với hình ảnh lá cờ đỏ sao vàng. Ngôi sao vàng năm cánh nhận được bằng cách sử dụng hàm fillpoly (10, poly10) để tô màu một đa giác có mười đỉnh là các điểm (x1, y1) ... (x10, y10) là các tọa độ của đỉnh ngôi sao. Tiết tấu nhanh, chậm của bản nhạc có thể thay đổi được bằng cách thay đổi giá trị của nốt đen d. Tiết tấu trung bình nhận được khi d = 300.

Chương trình có một số warning do chưa sử dụng hết một số định nghĩa về trường độ của nốt nhạc như các nốt móc kép, móc ba... Để tổng quát chúng tôi vẫn để các định nghĩa này tuy bài Tiến quân ca không dùng đến.

```
/* CT100 Chuong trinh sang tac nhac Bai Quoc ca */
#include<stdio.h>
#include<stdlib.h>
#include<graphics.h>
#include<dos.h>
/* Xac dinh cao do cua cac not nhac */
#define do1 66        /* do octave 1 */
#define dod1 70       /* do thang */
```

```
#define re1 73        /* re */
#define red1 78       /* re thang */
#define mi1 82        /* mi */
#define fa1 86        /* fa */
#define fad1 91       /* fa thang */
#define sol1 96       /* sol */
#define sold1 102     /* sol thang */
#define la1 108       /* la */
#define lad1 115      /* la thang */
#define si1 122       /* si */
#define do2 130       /* do octave 2 */
#define dod2 139      /* do thang */
#define re2 148       /* re */
#define red2 156      /* re thang */
#define mi2 164       /* mi */
#define fa2 176       /* fa */
#define fad2 188      /* fa thang */
#define sol2 196      /* sol */
#define sold2 209     /* sol thang */
#define la2 220       /* la */
#define lad2 233      /* la thang */
#define si2 247       /* si */
#define do3 264       /* do octave 3 */
#define dod3 281      /* do thang */
#define re3 297       /* re */
#define red3 313      /* re thang */
#define mi3 330       /* mi */
#define fa3 352       /* fa */
```

```
#define fad3 374     /* fa thang */
#define sol3 396     /* do thang */
#define sold3 415    /* sol */
#define la3 440      /* la */
#define lad3 468     /* la thang */
#define si3 495      /* si */
#define do4 528      /* do octave 4 */
#define dod4 565     /* do thang */
#define re4 594      /* re */
#define red4 625     /* re thang */
#define mi4 660      /* mi */
#define fa4 704      /* fa */
#define fad4 748     /* fa thang */
#define sol4 792     /* sol */
#define sold4 836    /* sol thang */
#define la4 880      /* la */
#define lad4 935     /* la thang */
#define si4 990      /* si */
#define do5 1050     /* do octave 5 */
#define lang 30000   /* dau lang la sieu am */

void not (unsigned ,float );     /* prototype not nhac */
int KhoiTaoDoHoa(void);          /* prototype do hoa */

void main()
{       /* Toa do hinh sao nam canh */
int poly10[] = { 320,150,285,225,200,225,270,270,240,350,
                 320,300,390,350,360,270,430,225,345,225 };
float d ,     /* not den */
```

```
tr,     /* not trang */
tro,    /* not tron */
md,     /* moc don */
mk,     /* moc kep */
m3,     /* moc ba */
m4,     /* moc bon */
dc,     /* not den cham */
trc,    /* not trang cham */
troc;   /* not tron cham */
printf("\n Vao gia tri not den d = ");scanf("%f",&d);
clrscr();
KhoiTaoDoHoa();
setbkcolor(RED);
setcolor(YELLOW);
setfillstyle(SOLID_FILL,YELLOW);
fillpoly(10,poly10);
/* Truong do cac not nhac */
tr = 2*d; tro = 4*d; md = d/2;
mk = d/4; m3 = d/8; m4 = d/16;
dc =3*d/2; trc =3*d;troc = 6*d;

/* Bat dau ban nhac TIEN QUAN CA */

not(re2,d);not(mi2,d);not(re2,d);not(sol2,tr);not(sol2,tro);
not(la2,d);not(sol2,d);not(si2,tr);not(si2,tro);not(la2,d);
not(sol2,d);not(mi2,tr);not(sol2,tr);not(sol2,d);not(mi2,d);
not(re2,d);not(si1,d);not(re2,tro);not(sol2,d);not(la2,d);
not(si2,tr);not(si2,tr);not(si2,tr);not(la2,d);not(sol2,d);
not(re3,tro);not(si2,d);not(sol2,d);not(la2,tr);not(la2,tr);
```

```
not(si2,tr);not(fad2,d);not(re2.d);not(sol2,tro);not(si2,d);
not(do3,d);not(re3,tr);not(re3,tr);not(mi3,tro);not(re3,d);
not(si2,tro);not(si2.tr);not(la2.d);not(sol2.tr);not(re2,tr);
not(fad2,tr);not(fad2,d);not(la2,d);not(sol2,tro);not(si2,d);
not(do3,d);not(re3,tr);not(re3,tr);not(mi3.tro);not(re3,d);
not(si2.tro);not(si2,tr);not(la2.d);not(sol2,tr); not(sol2,tr);
not(re2,tro);not(re3,tro);not(si2,tr);not(sol2,tr);not(mi3,tro);
not(re3,tr);not(si2,d);not(la2,d);not(re2,d);not(la2,tr);
not(la2,tro);not(si2,tr);not(sol2,tro);
}

void not(unsigned tanso, float dai) /* Ham xac dinh not nhac */
{
sound(tanso);
delay(dai);
nosound();
}

int KhoiTaoDoHoa(void) /* Ham khoi tao do hoa */
{
int graphdriver = DETECT,graphmode,error_code;
initgraph(&graphdriver,&graphmode, "..\\bgi");
error_code = graphresult();
if (error_code != grOk)
    {
    return(-1);
    if ((graphdriver != EGA ) && (graphdriver != VGA ))
        {
        closegraph();
```

```
        return(0);
        }
    return(1);
    }
}
```

Bài số 101 Tạo cửa sổ ở chế độ văn bản

Bài giải Để khởi tạo chế độ văn bản ta sử dụng :

```
textmode ( mode );
int mode;
```

để chọn một trong 5 chế độ sau đây :

Ký hiệu	*Giá trị số*	*Chế độ văn bản vidéo*
LASTMODE	-1	Chế độ văn bản có trước
BW40	0	Đen trắng 40 cột
C40	1	16 màu, 40 cột
BW80	2	Đen trắng 80 cột
C80	3	16 màu, 80 cột
MONO	7	Đơn sắc, 80 cột

Độc lập với chế độ văn bản được chọn màn hình luôn gồm 25 hàng

Đặt màu cửa sổ bằng `textbackground ( color );`
`int color;`

Đặt màu chữ dùng `textcolor ( color );`
`int color;`

Chương trình CT101 để kết thúc cuốn sách có cửa sổ gồm dòng chữ sau :

TRUONG DAI HOC BACH KHOA HA NOI

KHOA THIET BI DIEN

Cac tac gia chan thanh cam on ban doc

Neu ban co yeu cau chuong trinh xin moi

lien he voi KHOA THIET BI DIEN DT: 8692511

Chung toi xin dap ung mien phi

CAC TAC GIA

```
/* CT101 Chuong trinh lap cua so */
#include<stdio.h>
#include<conio.h>
void main()
{
textmode(C40) ; /* che do van ban 16 mau 40 cot */
/* Man hinh mau tim */
textbackground(MAGENTA) ;
window(1,1,40,25) ; /* Man hinh 25 X 80 */
clrscr() ;
textcolor(WHITE) ; /* Dong chu mau trang */
printf("\n\n TRUONG DAI HOC BACH KHOA HA NOI ") ;
printf("\n\n\n KHOA THIEI BI DIEN ") ;
printf("\n\n\n Cac tac gia chan thanh cam on ban doc ") ;
printf("\n\n Neu ban co yeu cau chuong trinh xin ") ;
printf("\n\n lien he KHOA THIET BI DIEN DT: 8692511 ") ;
printf("\n\n Chung toi xin dap ung mien phi ") ;
printf("\n\n CAC TAC GIA ") ;
getch();
return;
}
```

PHỤ LỤC

CÁC HÀM CỦA TURBO C

Hàm, thư mục	Cách sử dụng, ý nghĩa
abs math.h stdlib.h	int abs (int n) Trả về giá trị tuyệt đối của n
acos math.h	double acos (double x) Trả về arcos(x) nếu - 1 ≤ x ≤ 1, nếu không 0
asin math.h	double asin (double x) Trả về arsin(x) nếu - 1≤ x≤ 1, nếu không 0
atan math.h	double atan (double x) Trả về arctg(x)
atof math.h stdlib.h	double atof (const char * s) Chuyển xâu ký tự s thành một số dấu phẩy động
bioskey bios.h math.h	int bioskey (int n) Quản lý bàn phím bằng BIOS n = o đọc một phím n = 1 thả một ký tự đã sẵn sàng n = 2 tình trạng các phím đã bị thay đổi
calloc alloc.h stdlib.h	void * alloc (size_ t n, sixe_t t) Dành một khối nhớ nbx byte cho n đối tượng có kích thước t. Các byte của khối được khởi đầu bằng (\0)
ceil math.h	double ceil (double x) Làm tròn, trả về số nguyên nhỏ nhất lớn hơn hoặc bằng x
cleardevice graphics.h	void far cleardevice (void) Xóa màn hình đồ họa

clroel conio.h	void clroel (void) Xóa từ vị trí con trỏ đến cuối dòng trong cửa sổ hiện tại
clrscr conio.h	void clrscr (voil) Xóa cửa sổ hiện tại và đặt con trỏ về góc trái trên (tọa độ 1,1)
coreleft alloc.h	unsigned coreleft (void) unsigned long coreleft (void) Trả về kích thước bộ nhớ RAM không sử dụng. Kết quả kiểu unsigned long với mô hình tiny, small, medium và kiểu unisigned long với các mô hình compact, large và huge
cos math.h	double cos (double x) Trả về cos (x)
cosh math.h	double cosh (double x) Trả về cosh (x)
cprintf conio.h	int cprintf (const char *f [, arg, ...]) In các đối số theo các đặc tả
cputs conio.h	int cputs (const char *s) In xâu ký tự s trong cửa sổ văn bản hiện tại
exp math.h	double exp (double x) Trả về luỹ thừa cơ số tự nhiên của x
floor math.h	double floor (double x) Trả về trị nguyên lớn nhất nhỏ hơn hoặc bằng x
fabs math.h	double fabs (double x) Trả về giá trị tuyệt đối của số thực x
fillpoly graphics.h	void far fillpoly (int n, int far *tp) Vẽ và tô một đa giác n điểm có các tọa độ sắp xếp trong tp
free alloc.h stdtib.h	void free (void *b) Giải phóng khối nhớ b

ftime sys\timeb.h	void ftime (struct timeb *tb) Xác định ngày giờ hiện tại
getmaxcolor graphics.h	int far getmaxcolor (void) Lấy số màu cực đại trong chế độ đồ họa
getmaxx graphics.h	int far getmaxx (void) Lấy hoành độ cực đại của màn hình
getmaxy graphics.h	int far getmaxy (void) Lấy tung độ cực đại của màn hình
gettext conio.h	int gettext (int ax, int ay, int bx, int by, void *t) Sao chép một hình chữ nhật của màn hình văn bản vào vùng nhớ đệm do t trỏ vào
gettextinfo conio.h	void gettextinfo (struct text_info *r) Lấy thông tin về chế độ vidéo hiện thời
gotoxy conio.h	void gotoxy (int x, int y) Đặt con trỏ vào dòng y cột x trong cửa số
graphresult graphics.h	int far graphresult (void) Cho mã của lỗi đồ họa cuối cùng gặp phải
initgraph graphics.h	void far initgraph (&graphdriver,& graphmode," ") Khởi tạo hệ thống đồ họa, đặt card đồ họa, mode đồ họa với driver đã chọn
int86 dos.h	int int86 (int n, union REGS *ar,union REGS *br) Ngắt logic số n
isalnum ctype.h	int isalnum (int c) Trả về một giá trị khác không nếu c là một chữ cái hoặc ký tự số. Trái lại sẽ trả về 0
isdigit ctype.h	int isdigit (int c) Trả về một giá trị khác không nếu c là một con số từ 0 đến 9
kbhit conio.h	int kbhit (void) Thử gõ một phím trên bàn phím, sẵn sàng trên màn hình

line graphics.h	void far line (int ax, int ay, int bx, int by) Vẽ một đoạn thẳng nối hai điểm (ax,ay) và (bx,by) Vị trí con trỏ không thay đổi
linerel graphics.h	void far linerel (int dx, int dy) Vẽ một đoạn thẳng từ vị trí hiện tại (x,y) đến điểm (x+dx,y+dy). Con trỏ được di chuyển tới vị trí mới
lineto graphics.h	void far lineto (int nx, int ny) Vẽ một đoạn thẳng từ vị trí hiện tại (x,y) đến điểm (nx,ny). Con trỏ được di chuyển tới điểm (nx,ny)
log math.h	double log (double x) Trả về ln(x) , x > 0
log10 math.h	double log10 (double x) Trả về log(x) , x > 0
memmove mem.h	void *memmove (void *d, const void *s, size_t n) Sao chép một khối n byte của s vào trong d
memset mem.h	void *memset (void *s, int c, size_t n) Khởi đầu n byte của s với byte c
moveto graphics.h	void moveto (int nx, int ny) Chuyển vị trí hiện tại đến điểm có tọa độ (nx,ny)
outtextxy graphics.h	void outtextxy (int x, int y, char far *s) Ghi một xâu ký tự s vào điểm (x,y) trong cửa sổ đồ họa
pow math.h	double pow (double x, double y) Trả về x^y
putch conio.h	int putch (int c) Đưa một ký tự vào màn hình trong cửa sổ làm việc
puttext conio.h	int puttext (int ax, int ay, int bx, int by) Chép lại nội dung của vùng nhớ đệm t vào hình chữ nhật của màn hình văn bản
random stdlib.h	int random (int n) Phát một số nguyên ngẫu nhiên giữa 0 và n-1
radommize stdlib.h	void randomize (void) Khởi đầu bộ sinh số ngẫu nhiên

realloc stdlib.h alloc.h	void *realloc (void *b, size_t n) Điều chỉnh kích thước của khối b theo kích thước n
rectangle graphics.h	void far rectangle (int ax, int ay, int bx, int by) Vẽ một hình chữ nhật có góc trái trên (ax,ay) và góc phải dưới (bx,by)
setbkcolor graphics.h	void far setbkcolor (int c) Chọn màu nền cho màn hình đồ họa
setcolor graphics.h	void far setcolor (int c) Chọn màu vẽ
setfillstyle graphics.h	void far setfillstyle (int motif, int c) Đặt mẫu tô và màu tô đa giác
sin math.h	double sin (double x) Trả về sin(x)
sinh math.h	double sinh (int x) Trả về sh(x)
sprintf stdio.h	int sprintf (char *b, const char *f [, arg, ...]) Ghi các đối số trong xâu ký tự b theo đặc tả f
sqrt math.h	double sqrt (double x) Trả về căn bậc hai của x
strcat string.h	char *strcat (char *d, const char *s) Sao chép xâu nguồn s vào xâu nhận d
strchr string.h	char *strchr (const char *s, int c) Tìm lần xuất hiện của ký tự c trong xâu s
strcmp string.h	int strcmp (const char *sa, const char *sb) So sánh xâu sa và sb. Trả về 0 nếu các xâu giống nhau, trả về giá trị âm nếu sa nhỏ hơn sb và giá trị dương nếu sa lớn hơn sb
strcpy string.h	char *strcpy (char *d, const char *s) Sao chép xâu nguồn s vào cuối xâu nhận d
strdup string.h	char *strdup (const char *s) Dành vị trí và sao chép xâu tạo nên bản sao

strlen string.h	size_t strlen (const char *s) Trả về độ dài của xâu s, không tính đến ký tự không
strncat string.h	char *strncat (char *d, const char *s, size_t n) Móc nối n byte của xâu nguồn s vào xâu nhận d
strncpy string.h	char *strncpy (char *d, const char *s, size_t l) Sao chép l byte của xâu nguồn s vào xâu nhận d bằng cách cắt d hay thêm các ký tự không
strnset string.h	char *strnset (char *s, int c, size_t n) Khởi đầu n ký tự đầu của xâu s với ký tự c
strtod stdlib.h	double *strtod (const char *s, char **p) Chuyển xâu s thành giá trị kiểu double, p trỏ về ký tự đầu tiên gây dừng
tan math.h	double tan (double x) Trả về tg(x)
toupper ctype.h	int toupper (int c) Chuyển chữ thường từ a đến z thành chữ hoa
va_arg stdarg.h	type va_arg (va_liste pvar, type) Đối số của danh sách thông số biến thiên pvar là kiểu type
va_end stdarg.h	void va_end (va_liste pvar) Dành cho lối ra bình thường của một hàm đối số thay đổi
va_start stdarg.h	void va_start (va_liste pvar, pfixe) Đặt vị trí pvar vào đối số thay đổi đầu tiên sau đối số cố định cuối cùng pfixe
wherex conio.h	int wherex (void) Trả về cột của con trỏ trong cửa sổ làm việc
wherey conio.h	int wherey (void) Trả về dòng của con trỏ trong cửa sổ làm việc
window conio.h	void window (int ax, int ay, int bx, int by) Định cửa sổ văn bản ở góc trái trên tọa độ (ax,ay) đến góc phải dưới tọa độ (bx,by)

TÀI LIỆU THAM KHẢO

[1] 101 thuật toán và chương trình bài toán khoa học và kinh tế ngôn ngữ PASCAL, Lê Văn Doanh - Trần Khắc Tuấn, NXB KHKT 1993

[2] Exercices en Langage C. Claude Delannoy, Eyrolles 1992

[3] TURBO C, G. Leblance, Eyrolles 1990

[4] Outils mathématique en C++, Alain Reverchon, Dunod 1993

[5] TURBO C des outils, Albert Levine, Berti Editions 1993

[6] TURBO C++ Bible, Nabaiyoti - Barkakati, The Waite Group's UNIX 1991

[7] The C proramming language, B. Kernighan and D. Ritchie, Prentice Hall 1989

[8] TURBO C, Richard Winner (Bản tiếng Nga) 1992

[9] Programmer's guide Borland C++ Version 4.0, Borland International, Inc 1993

LÊ VĂN DOANH TRẦN KHẮC TUẤN
LÊ ĐÌNH ANH

101 THUẬT TOÁN VÀ CHƯƠNG TRÌNH BÀI TOÁN KHOA HỌC KỸ THUẬT VÀ KINH TẾ BẰNG NGÔN NGỮ C

Chịu trách nhiệm xuất bản : Pgs.Pts. TÔ ĐĂNG HẢI

Biên tập : HỒNG VŨ

Sửa bản in : XUÂN PHƯƠNG

Vẽ bìa : TRẦN THẮNG

NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT
HÀ NỘI – 2006

In 500 cuốn, khổ 14,5 x 20,5 cm, tại Xưởng in NXB Văn hoá dân tộc.
Giấy phép xuất bản số: 546 – 40 – 6 / 10 / 2005.
In xong và nộp lưu chiểu tháng 12 / 2005.

205339



Giá: 30.000đ